

Universitat Oberta de Catalunya

# MEMÒRIA

## SISTEMA DE GESTIÓ D'AMONESTACIONS I SANCIONS EN CENTRES EDUCATIUS

Raquel Cosano Puerta

Consultor: Jordi Ferrer Duran



Primavera  
2011

Índex

Resum del projecte .....	5
1- Introducció a l'enunciat .....	6
1.1 Objectiu i justificació del projecte .....	6
1.2 Enfocament i mètode seguit .....	6
1.3 Requeriments del projecte .....	6
1.4 Planificació del projecte .....	9
1.5 Introducció al sistema de treball .....	12
2- Disseny de la base de dades .....	13
2.1 Disseny conceptual .....	13
2.1.1 Introducció .....	13
2.1.2 Entitats i atributs .....	13
2.1.2.1 Entitat PROFESSOR .....	13
2.1.2.2 Entitat ALUMNE .....	14
2.1.2.3 Entitat TUTOR .....	15
2.1.2.4 Entitat TIPUS AMONESTACIÓ .....	16
2.1.2.5 Entitat AMONESTACIÓ .....	16
2.1.2.6 Entitat TIPUS SANCIO .....	17
2.1.2.7 Entitat SANCIO .....	18
2.1.2.8 Entitat ASSIGNATURA .....	18
2.1.2.9 Entitat CALENDARI .....	19
2.1.2.10 Entitat CURS .....	19
2.1.2.11 Entitat MATRICULACIÓ .....	20
2.1.2.12 Entitat DOCENCIA .....	20
2.1.2.13 Sistema de control .....	21
2.1.3 Diagrama del model entitat relació .....	21
3- Disseny físic del projecte .....	25

3.1 Introducció: Transformació al model relacional .....	25
3.2 Creació dels scripts i procedures .....	26
3.2.1 Creació de les taules .....	26
3.2.1.1 Taula PROFESSOR .....	26
3.2.1.2 Taula ALUMNE .....	26
3.2.1.3 Taula TUTOR .....	26
3.2.1.4 Taula TIPUSAMONESTACIO .....	27
3.2.1.5 Taula AMONESTACIO .....	27
3.2.1.6 Taula TIPUSSANCIO .....	27
3.2.1.7 Taula SANCIO.....	27
3.2.1.8 Taula ASSIGNATURA .....	28
3.2.1.9 Taula CALENDARI .....	28
3.2.1.10 Taula CURS .....	28
3.2.1.11 Taula MATRICULACIO .....	28
3.2.1.12 Taula DOCENCIA .....	29
3.2.1.13 Taula FESTIU .....	29
3.2.1.14 Taula LOG .....	29
3.2.1.15 Taules estadístiques .....	29
3.2.2 Creació dels procediments d'alta, de baixa i de modificació .....	32
3.2.2.1 PROFESSOR .....	32
3.2.2.2 ALUMNE .....	33
3.2.2.3 CURS .....	34
3.2.2.4 ASSIGNATURA .....	35
3.2.2.5 MATRICULACIO .....	36
3.2.2.6 CALENDARI .....	37
3.2.2.7 TIPUSAMONESTACIO .....	38
3.2.2.8 TIPUSSANCIO .....	39

3.2.2.9 AMONESTACIO .....	41
3.2.2.10 SANCIO .....	42
3.2.2.11 TUTOR .....	42
3.2.2.12 DOCÈNCIA .....	43
3.2.3 Creació dels llistats requerits .....	43
3.2.3.1 Amonestacions imposades .....	43
3.2.3.2 Alumnes d'un curs .....	44
3.2.3.3 Amonestacions i sancions disponibles en el centre educatiu .....	44
3.2.3.4 Amonestacions per alumne .....	44
3.2.3.5 Sancions per alumne i any .....	44
3.2.3.6 Amonestacions per professor i any .....	45
3.2.3.7 Sancions per a curs i any .....	45
3.2.3.8 Alumne més sancionat en un any .....	45
3.2.3.9 Professor més amonestador per curs .....	45
3.2.3.10 Mitja de les sancions que tenen els alumnes per curs .....	46
3.2.3.11 Alumnes que no tenen cap amonestació .....	46
3.2.3.12 Llistat de curs i alumne .....	46
3.2.3.13 Llistat de professor i assignatura .....	46
3.2.3.14 Llistat del professor responsable de cada curs .....	47
3.2.3.15 Llistat de les hores de consulta als alumnes .....	47
3.2.3.16 Llistat de les hores de consulta als pares .....	47
3.2.3.17 Llistat de les amonestacions i sancions d'un alumne .....	47
3.3 Creació del joc de proves .....	48
4. Valoració econòmica del projecte .....	52
5. Conclusions .....	54
6. Glossari .....	55
7. Bibliografia .....	56

## **Resum del projecte**

El treball que es presentarà a continuació ha consistit en la implementació i l'elaboració del disseny d'un sistema que permet la gestió d'amonestacions i sancions en els centres educatius.

El treball consta de varies parts diferenciades. Primerament és parlarà dels objectius a adquirir i dels requeriments imposats per la Generalitat de Catalunya, en aquest mateix apartat es disposaran els mètodes i la planificació a seguir per tal de complir els requisits.

Posteriorment s'exposarà el disseny conceptual basant-se en un esquema entitat-relació que permetrà localitzar les diferents taules, camps, tipus de dades i les restriccions que contindran les referències introduïdes a la base de dades. Tanmateix s'obtindrà el diagrama entitat-relació.

Un cop definides les entitats, es procedirà al disseny físic de la base de dades on s'exposaran els procediments que s'encarregaran de crear, eliminar, modificar o actualitzar les dades de la base de dades.

Per últim, s'han generat una sèrie de dades fictícies d'un centre ubicat a Catalunya. Amb aquestes dades i amb l'ajuda del joc de proves elaborat es podrà comprovar l'efectivitat i la robustesa que tindrà la base de dades.

## **1. Introducció a l'enunciat**

### **1.1 Objectiu i justificació del projecte**

En el següent treball es tractarà de donar una proposta per a un disseny i implementació d'un sistema d'amonestacions i sancions en un centre educatiu d'ensenyament secundari per tal que la Generalitat de Catalunya el pugui implementar en cada un dels seus centres educatius.

Els objectius a cobrir en el treball seran tots aquells especificats en la demanda de la Generalitat de Catalunya.

Els objectius personals són els d'aprofundir en el llenguatge SQL, assolir els conceptes necessaris del gestor de bases de dades Oracle 10g i els d'aplicar els coneixements adquirits durant la carrera, en aquest cas Enginyeria de Telecomunicacions especialitat en Telemàtica, així com ampliar-los utilitzant les noves eines.

### **1.2 Enfocament i mètode seguit**

Per tal de donar una proposta es crearà, mitjançant el programari d'Oracle 10g Express Edition i el SQLDeveloper, les estructures de dades i mètodes necessaris per tal de poder accedir a les dades introduïdes.

No obstant, es dona per suposat que la Generalitat de Catalunya disposa d'una intranet de propòsit múltiple, realitzada a mida i que ha anat evolucionant amb diferents desenvolupadors quan ha necessitat noves propostes. En aquest cas, per tant, no contemplarem ni el disseny de la interfície d'usuari ni el programari d'instal·lació del servidor ni del client.

### **1.3 Requeriments del projecte**

El sistema a dissenyar ha de permetre emmagatzemar tota la informació de l'alumnat matriculat, dels cursos on estan matriculats i dels seus professors així com també generar les consultes que es realitzin.

Així doncs aquest sistema també s'encarregarà de pre-calcular i emmagatzemar diversa informació estadística segons els requisits establerts.

Els requisits i les funcionalitats que s'establiran a la base de dades s'exposen a continuació:

- Ha de permetre introduir i mantenir les dades bàsiques dels alumnes i professors del centre corresponent.
- Ha de permetre introduir i mantenir els diferents cursos que s'imparteixin anualment i les assignatures de cadascun d'ells.
- Ha de possibilitar la introducció dels calendaris escolars en relació als cursos definits.
- Ha de permetre definir amonestacions que seran establertes pels professors. També ha de facilitar la definició de nous tipus d'amonestacions.
- Ha de permetre introduir una sanció indicant els motius i la resolució. Aquestes seran imposades per una acumulació d'amonestacions o una amonestació greu.

- Ha de permetre definir noves regles o modificar les existents.
- Ha de permetre saber en quin curs pertany cada alumne, quins professors donen cada assignatura i quins professors són responsables de cada curs.
- Ha de permetre emmagatzemar l'horari d'atenció del professor respecte els alumnes i els pares.
- Ha de permetre conèixer quines amonestacions i sancions té en cada moment cadascun dels alumnes.

Per tal d'efectuar això es crearan els següents procediments:

- Procediment ABM\* dels alumnes.
- Procediment ABM dels professors.
- Procediment ABM dels cursos.
- Procediment ABM de les assignatures.
- Procediment ABM del calendari escolar.
- Procediment ABM de les amonestacions i sancions (tenint en compte les noves tipologies).
- Procediments de consulta (l·listat de totes les amonestacions, l·listat de tots els alumnes d'un curs, l·listat de tots tipus d'amonestacions i sancions disponibles en el centre, l·listat de totes les amonestacions i sancions d'un alumne).

Com a funcionalitat també haurem d'implementar un mòdul estadístic per donar resposta a les següents consultes:

- Número d'amonestacions per alumne.
- Número de sancions per alumne i any.
- Mitjana d'amonestacions per professor i any.
- Número de sancions per curs i any.
- Nom dels alumnes més sancionats en un any donat.
- Nom del professor que posa més amonestacions per curs.
- Mitjana de sancions que tenen els alumnes per curs.
- Número d'alumnes que no tenen cap amonestació.

**\*(ABM: Alta + Baixa + Modificació')**

Per a poder dur a terme aquests requeriment, s'assignaran els recursos següents:

- Per part nostra:

Recursos	Roles
Cap de projecte	S'encarregarà de coordinar tècnicament el projecte i d'elaborar el disseny global del sistema. Responsable de l'anàlisi funcional i la estimació del temps. Raquel Cosano ( <a href="mailto:rcosanop@uoc.edu">rcosanop@uoc.edu</a> )
Desenvolupador especialitzat	Desenvolupador especialitzat en la implementació de bases de dades mitjançant Oracle Express v10.2.0.1. Raquel Cosano ( <a href="mailto:rcosanop@uoc.edu">rcosanop@uoc.edu</a> )
Testing	Encarregat de testejar i validar cada una de les fases del desenvolupament. Raquel Cosano ( <a href="mailto:rcosanop@uoc.edu">rcosanop@uoc.edu</a> )

- Per part de la Generalitat de Catalunya:

Recursos	Roles
Cap de projecte	Serà la persona amb capacitat de decisió que coordinarà la toma de requeriments per part del client i serà el responsable de la interlocució amb l'equip de treball. Efectuarà les validacions en les reunions.
Usuari clau	Responsable a nivell funcional de que la definició del sistema respon als requeriments definits.

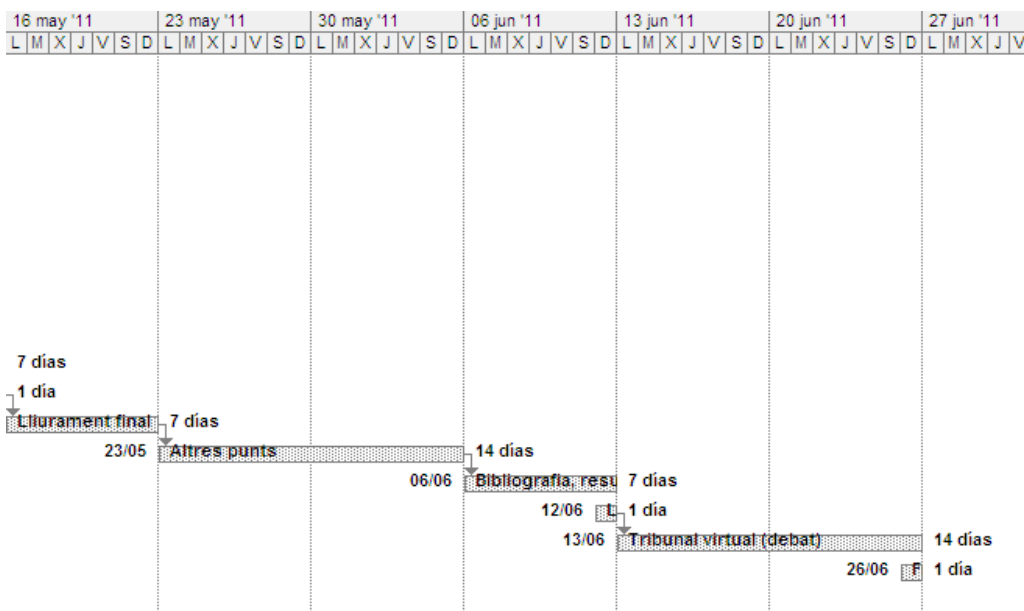
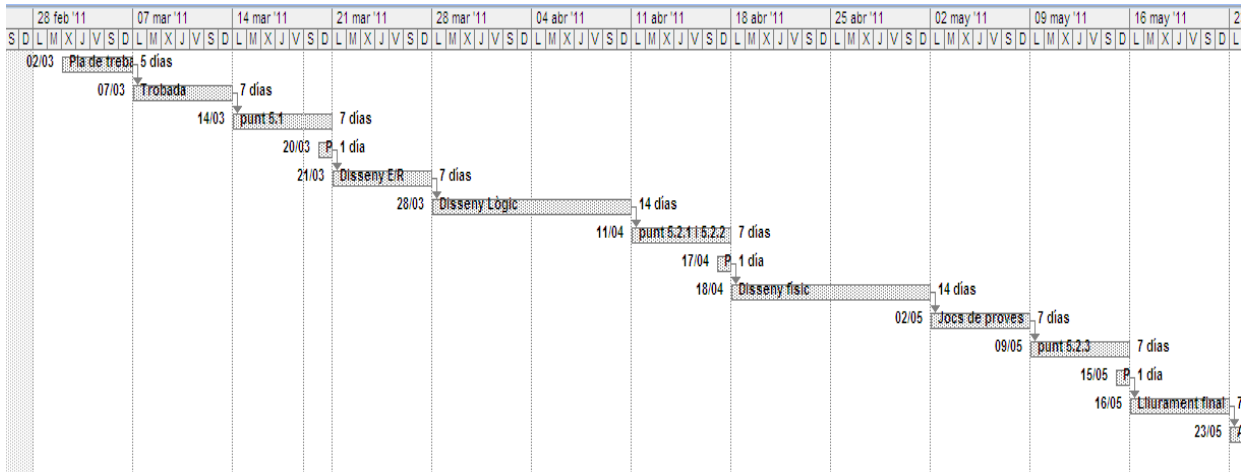


### 1.4 Planificació del projecte

A continuació es mostra el diagrama amb el temps aproximat per a la realització del projecte. En total es disposa de 15 setmanes per finalitzar la presentació, les quals han sigut disposades de la següent manera:

Setmana	Dates	Activitat	Esdeveniment
1	2 a 6 de Març	Presentació i pla de treball	Presentació
2	7 a 13 de Març	Presentació i pla de treball	Trobada presencial: dissabte 12 març
3	14 a 20 de Març	Redacció del punt 5.1 de la memòria	Lliurament del pla de treball: diumenge 20 de març
4	21 a 27 de Març	Disseny del model E/R	
5	28 de Març a 3 de Abril	Disseny lògic	
6	4 a 10 de Abril	Implementació del disseny lògic	
7	11 a 17 de Abril	Redacció del punt 5.2.1 i 5.2.2 de la memòria	Lliurament de la PAC2: diumenge 17 de abril
8	18 a 24 de Abril	Disseny dels scripts i procedures	
9	25 de Abril a 1 de Maig	Implementació dels scripts i procedures	
10	2 a 8 de Maig	Comprovació jocs de proves	
11	9 a 15 de Maig	Redacció del punt 5.2.3 de la memòria	Lliurament de la PAC3: diumenge 15 de maig
12	16 a 22 de Maig	Inici del lliurament final (Memòria + Presentació +Producte)	
13	23 a 29 de Maig	Redacció dels altres punts de la memòria	
14	30 de Maig a 5 de Juny	Redacció dels altres punts de la memòria	
15	6 a 12 de Juny	Últims retocs (Bibliografia, resum i comentaris)	Lliurament final (Memòria + Presentació +Producte): diumenge 12 de juny
16	13 a 19 de Juny	-	
17	20 a 26 de Juny	Inici debat Tribunal Virtual: 20 de juny	Final debat del Tribunal Virtual: 26 ed juny

		Nombre de tarea	Duración	Comienzo	Fin
1		Presentació Pla de treball	5 días	mié 02/03/11	dom 06/03/11
2		Trobada presencial	7 días	lun 07/03/11	dom 13/03/11
3		Redacció del punt 5.1 de la memòria	7 días	lun 14/03/11	dom 20/03/11
4		Lliurament del Pla de Treball	1 día	dom 20/03/11	dom 20/03/11
5		Disseny del model E/R	7 días	lun 21/03/11	dom 27/03/11
6		Disseny lògic implementació	14 días	lun 28/03/11	dom 10/04/11
7		Redacció del punt 5.2.1 i 5.2.2 de la memòria	7 días	lun 11/04/11	dom 17/04/11
8		Lliurament de la PAC2	1 día	dom 17/04/11	dom 17/04/11
9		Disseny i implementació dels scripts i procedures	14 días	lun 18/04/11	dom 01/05/11
10		Jocs de proves	7 días	lun 02/05/11	dom 08/05/11
11		Redacció del punt 5.2.3 de la memòria	7 días	lun 09/05/11	dom 15/05/11
12		Lliurament de la PAC3	1 día	dom 15/05/11	dom 15/05/11
13		Inici del lliurament final	7 días	lun 16/05/11	dom 22/05/11
14		Redacció dels altres punts de la memòria	14 días	lun 23/05/11	dom 05/06/11
15		Bibliografia, resum i comentaris	7 días	lun 06/06/11	dom 12/06/11
16		Lliurament final	1 día	dom 12/06/11	dom 12/06/11
17		Tribunal virtual (debat)	14 días	lun 13/06/11	dom 26/06/11
18		Final Tribunal virtual (debat)	1 día	dom 26/06/11	dom 26/06/11



Les tasques a realitzar es defineixen a continuació amb més detall:

1- *Instal·lació del programari a utilitzar.*

Per a fer aquest disseny i posterior implementació s'utilitzarà el gestor de bases de dades Oracle 10g Express , eines de disseny de BD (SQLDataModeler) i una eina d'editor SQLdeveloper.

Aquesta tasca és la més important ja que s'ha de tenir cura de solucionar tots els possibles problemes que ens puguin sorgir al llarg de la instal·lació. En aquesta tasca també s'inclou la familiarització amb les eines instal·lades per tal de conèixer totes les possibilitats que ens poden oferir aquestes eines.

La instal·lació del programari va ser una tasca ràpida i eficient sense gaires problemes. En canvi, la familiarització amb aquest programari nou va ser força dura. Per tal de poder utilitzar-lo amb eficiència s'han tingut que llegir uns quants tutorials.

2- *Disseny del model lògic.*

En aquesta tasca es farà la definició dels elements, entitats i les relacions necessàries pel projecte a realitzar.

Es farà el disseny lògic utilitzant SQLDataModeler.

Els objectius d'aquesta tasca són crear un conjunt d'entitats, les quals han de ser un exemple clar de com s'estructuren els centres educatius. S'ha generat un ordre de prioritats que han sigut escollides segons els requeriments, per després començar per introduir les entitats més obvies i les seves relacions. Amb això s'ha pogut establir un disseny lògic, que després en crear els procediments (tasca 3) ha tingut que ser un altre cop estructurat (s'han suprimit entitats que no feien falta i unes altres s'han enfocat d'una altra manera).

3- *Implementació d'aquest disseny mitjançant Oracle BD Express.*

En aquesta tasca es definiran les taules que es crearan gràcies al model relacional de la tasca segona.

La creació de les taules, un cop definides les entitats en la tasca anterior, ha sigut la feina més fàcil de tot el projecte. En aquest punt ja tenia en ment tota l'estructura i només quedava introduir-la amb la base de dades.

4- *Disseny dels scripts i procedures.*

En aquesta tasca es farà la definició dels procedures amb els seus paràmetres d'entrada i sortida.

Es realitzaran les diferents funcionalitats demanades al començament del projecte. Amb aquesta tasca es definiran procedures tals com donar d'alta, de baixa o procedures de modificació de totes les taules abans descrites.

El disseny dels procedures ha estat una tasca molt difícil. Hi ha hagut procediments que s'han pensat ràpid i han estat a la primera ben fets com alta d'un alumne o professor, però d'altres com l'alta d'una amonestació o com el mòdul estadístic han sigut molt més complicades.

En aquest punt la planificació del temps que s'havia establert en un principi ha fallat perquè s'han hagut d'utilitzar uns quants dies més per tal de que tots els procediments estiguessin acabats i funcionant correctament.

5- *Generar les dades de la base de dades*

Per tal de que la base de dades es converteixi en un cas real, es crearà un conjunt de dades que seran d'ajuda per confirmar que la base de dades funciona de manera correcta.

6- *Generar el joc de proves.*

En aquesta tasca es farà ús d'un joc de proves per tal de, a nivell funcional, determinar si el sistema respon als requeriments definits pel client (Generalitat de Catalunya).

Aquest joc de proves s'ha definit per realitzar totes les possibilitats que amb la base de dades es poden crear.

Una de les tasques més importants i més interessants ha estat aquesta, no va ser fàcil tenir el joc de proves ni molt menys fer-lo anar correctament però, si més no, ha estat una feina molt important per tot el seu significat (que la BD ha sigut estructurada de manera correcta).

A més, el joc de proves és una de les claus d'aquest projecte i per aquest fet s'ha de dir, que es va trigat una setmana més del previst prèviament.

7- *Redacció dels capítols*

Aquesta tasca, la més important, consta de la descripció detallada de tots els capítols que estaran definits en la memòria final.

Aquesta tasca es durà a terme durant tot el desenvolupament del projecte (les 15 setmanes) paral·lelament a les altres tasques.

No és gens fàcil escriure tot allò que ha estat present en tot el projecte i molt menys tot el que s'ha pogut plantejar abans d'arribar a l'estructura que es creu que és la millor.

## 1.5 Introducció al sistema de treball

S'han hagut de plantejar algunes qüestions prèvies amb la finalitat de definir com s'estructuren els centres educatius. Per aquest motiu, a continuació s'exposarà l'estructura d'aquests.

Principalment hi haurà un únic centre educatiu (després podrà ser implantat en més d'un centre).

Aquest centre tindrà contractats a un determinar nombre de professors. Tots ells faran unes determinades tasques, ja sigui com a professor responsable d'alguna assignatura o com a tutor d'alguns dels alumnes que estaran matriculats al centre.

El cas de direcció del centre ara no és prioritari ja que l'objectiu és el de crear un sistema gestor d'amonestacions i sancions que podran imposar els professors als alumnes.

Els alumnes hauran d'estar matriculats en algun dels cursos que imparteix el centre. Aquests cursos tindran un horari propi determinat, a més, en els cursos s'impartiran assignatures amb un dels professors.

Tenint en compte tot això, entra en escena el concepte de docència. S'anomena docència a la relació que hi ha entre curs, assignatura i professor, o dit d'una altra manera, la docència seria com l'anomenada "classe" en el vocabulari dels alumnes.

Per finalitzar, l'objectiu del sistema és crear dintre de tota aquesta estructura un sistema que ens indiqui els tipus d'amonestacions i sancions i el nombre de totes les amonestacions i de totes les sancions que es poden imposar al llarg d'un any acadèmic.

## 2. Disseny de la base de dades

### 2.1 Disseny conceptual

#### 2.1.1 Introducció

En un disseny d'una base de dades s'estableixen diferents etapes, cadascuna és un resultat intermedi que serveix de punt de partida de l'etapa següent.

Aquestes etapes se les coneix com a disseny conceptual i el disseny físic.

En aquest punt obtindrem el disseny conceptual, on es troba l'estructura de la informació de la base de dades expressada en un model entitat-relació que permet representar l'esquema conceptual d'una base de dades de forma gràfica mitjançant el diagrama E/R\*.

Primer es consideren tots els requeriments abans esmentats per tal de construir les entitats, atributs i interrelacions necessaris i, per últim, es veurà el diagrama E/R.

#### 2.1.2 Entitats i atributs

L'entitat és qualsevol objecte del món real amb existència pròpia sobre el qual es té informació en una base de dades. Els atributs representen la definició d'aquestes entitats.

A continuació es passa a definir les entitats necessàries per crear la base de dades.

Les claus primàries seran expressades amb Pk i les claus foranes seran expressades amb Fk.

##### 2.1.2.1 Entitat PROFESSOR

Professor contindrà informació relativa a tots els professor que estan en el centre (donin o no classe). Emmagatzema un historial de tots els professor que han passat pel centre.

La taula PROFESSOR conté un ID únic per a cada individu i un camp anomenat ESTAT que indica si es actiu, donant classe al institut, o ja no hi és. Aquest camp es un char(1) amb dos valors possibles:

N – Ja no hi és al institut

S – Es actiu al institut i donant classe.

Si un professor que ja ha passat pel centre tornés, caldria modificar només aquest camp de N a S per tenir-lo de nou actiu.

Els altres atributs són informatius requerits pel client. Cal destacar els atributs atenció als pares i alumnes que són d'obligatòria introducció.

\* *E/R: entitat - relació*

Els atributs associats seran:

- IdProfessor (Pk), clau primària de l'entitat PROFESSOR. Es defineix com el document identificatiu de cada professor, el seu DNI o NIF. CHAR(255).
- Nom, nom de professor. CHAR(100).
- Cognom1, primer cognom del professor. CHAR(100).
- Cognom2, segon cognom del professor. CHAR(100).
- Telèfon, telèfon personal del professor. CHAR(100).
- Direcció, adreça on viu el professor. CHAR(250).
- Població, lloc on està situada la vivenda del professor. CHAR(250).
- Província, província de la vivenda on viu el professor. CHAR(250).
- País, país on resideix el professor. CHAR(250).
- Email, direcció de correu del professor. CHAR(250).
- Edat, edat del professor. NUMBER(10,0).
- Horari atenció pares, horari que té el professor per parlar amb els pares dels alumnes. CHAR(100).
- Horari atenció alumnes, horari que té el professor per parlar amb els alumnes. CHAR(100).
- Estat, aquest atribut ens indicarà si el professor esta actiu, o sigui, si està donant classe a l'institut o ja no hi és.

### 2.1.2.2 Entitat ALUMNE

Alumne contindrà informació relativa a tots els alumnes que estan o han estat inscrits en el centre.

ALUMNE disposa d'un ID únic per tal de diferenciar als alumnes amb una clau principal. Aquesta clau s'ha pensat que pot ser el DNI de l'alumne. D'aquesta manera ens estalviem crear una clau única, si podem fer servir la dada DNI obligatòria com a identificador únic.

Aquesta clau serà del tipus CHAR i contindrà el DNI o NIE dels alumnes que estan o han passat pel centre. Es tindrà en compte, que aquest sistema pot generar un "bug" en el següent escenari:

"Un alumne estranger amb una identificació NIE, es nacionalitza i s'identifica ara amb el DNI". En aquest hipotètic cas minoritari, el sistema crearia un nou alumne diferent fruit de la nova identificació estatal de l'alumne.

Com passa a la taula PROFESSOR, aquesta taula també conté un camp denominat ESTAT que guarda la informació de la presència o no d'un alumne al centre.

Els altres camps són per indicacions informatives del client. On es podrà trobar molta informació d'un alumne, on molts són opcionals com el nom del pare o mare, ja que pot donar-se el cas de que sigui orfe de pare o mare. En el cas de les adreces dels pares, si aquests estan separats es tindria que donar l'opció d'indicar dues adreces, es per aquest motiu que disposa de dos atributs adreça, un per cada progenitor.

Els atributs associats seran:

- IdAlumne (Pk), clau primària de l'entitat ALUMNE. Es defineix com el document identificatiu de cada alumne, el seu DNI o NIF. CHAR(255).
- Nom, nom de l'alumne. CHAR(100).
- Cognom1, primer cognom de l'alumne. CHAR(100).
- Cognom2, segon cognom de l'alumne. CHAR(100).
- Telèfon, telèfon personal de l'alumne. CHAR(100).
- Direcció, adreça on viu l'alumne. CHAR(250).
- Població, lloc on està situada la vivenda on viu l'alumne. CHAR(250).
- Província, província de la vivenda on viu l'alumne. CHAR(250).
- País, país on resideix l'alumne. CHAR(250).
- Email, direcció de correu de l'alumne. CHAR(250).
- Edat, edat que té l'alumne. NUMBER(10,0).
- Nom pare, representa el nom del pare de l'alumne. CHAR(255).
- Nom mare, representa el nom de la mare de l'alumne. CHAR(255).
- Direcció pare, representa l'adreça on viu el pare de l'alumne.
- Direcció mare, representa l'adreça on viu la mare de l'alumne.
- Telèfon pare, representa el telèfon de contacte del pare de l'alumne. CHAR(255).
- Telèfon mare, representa el telèfon de contacte de la mare de l'alumne. CHAR(255).
- Estat, aquest atribut, com passa a l'atribut estat de professor, ens indicarà si un alumne està actiu o no, o sigui si està inscrit en algun curs.

### 2.1.2.3 Entitat TUTOR

Tutor contindrà informació relativa a tots els TUTORS i indicarà el tutor que té en cada moment cada alumne.

TUTOR té dos claus foranes en els atributs IDALUMNE i IDPROFESSOR que fan referència a les taules ALUMNE i PROFESSOR respectivament.

La clau primària d'aquesta taula és IDALUMNE. Aquesta clau indica que un alumne només pot tenir assignat a un professor com a tutor. A més cada alumne té que tenir assignat el seu tutor, aquest però podrà ser substituït per un altre.

S'ha considerat que el tutor no té cap relació amb el curs, o sigui, en un mateix curs no tots els alumnes obligatòriament tindran el mateix tutor, però es procurarà que sigui així. L'estructura ens permetrà llavors, que l'alumne canviï de tutor tal i com passa a la UOC.

Els atributs associats seran:

- IdProfessor(Fk), clau forana de l'entitat PROFESSOR. Identifica als professor que arriben a ser tutors d'algun alumne. CHAR(255).
- IdAlumne(Pk, Fk), clau forana de l'entitat ALUMNE. Identifica als alumnes que tenen un tutor assignat. CHAR(255).

#### 2.1.2.4 Entitat TIPUS AMONESTACIÓ

Tipus amonestació contindrà la informació relativa als tipus d'amonestacions que pot posar un professor a un alumne en el centre escolar.

TIPUSAMONESTACIO té el camp IDTIPUSAMONESTACIO que ens identificarà com a clau primària els tipus d'amonestacions que es poden posar en el centre.

El camp descripció descriurà la amonestació en un llenguatge humà.

Com passa a la taula PROFESSOR, aquesta taula també conté un camp denominat ESTAT que guarda la informació de la presència o no d'un tipus d'amonestació al centre.

Així doncs, aquesta taula contindrà de manera predefinida les amonestacions requerides en un primer moment pel client. De totes maneres, aquestes podran ser modificades o bé tal i com demanen les especificacions, es podran afegir nous tipus d'amonestacions fàcilment.

Tipus amonestació	Descripció
alumne_arriba_tard	Quan l'alumne arriba tard a classe
alumne_soroll	L'alumne fa molt de soroll en classe
Alumne_malparlat	L'alumne és malparlat
Alumne_no_deures	L'alumne no fa els deures

Els atributs associats seran:

- IdTipusamonestacio (Pk), identificarà els tipus d'amonestacions que es poden imposar en el centre. NUMBER(10,0).
- Descripció, aquest atribut descriurà l'amonestació en un llenguatge humà. CHAR(250).
- Estat, aquest atribut ens indicarà si el tipus d'amonestació esta actiu, o sigui, si es pot imposar amonestació per aquesta raó o no.

#### 2.1.2.5 Entitat AMONESTACIÓ

Amonestació contindrà informació relativa a totes les amonestacions que s'han imposat al centre. AMONESTACIÓ té un IDAMONESTACIO que farà que cada una de les amonestacions introduïdes sigui única. Els atributs són de tipus NUMBER, CHAR i DATE que tenen valor NOT NULL, és a dir no podran contenir valors nuls.

L'únic atribut que pot contenir valors nuls és el camp COMUNICATPARE. Aquest s'emplena quan l'amonestació s'ha comunicat als pares o no.

Es pot veure que té tres claus foranes que identifiquen al professor, a l'alumne i al tipus d'amonestació per tal de definir qui ha fet l'amonestació (professor), qui l'ha rebut (alumne) i de quin tipus és (tipus amonestació).



Els atributs associats a ella seran:

- IdAmonestacio (Pk), identificarà a cada amonestació imposada per qualsevol professor a qualsevol alumne de qualsevol tipus. NUMBER(10,0).
- IdAlumne (Fk), relaciona cada amonestació amb un alumne en concret. CHAR(255).
- IdProfessor (Fk), relaciona cada amonestació amb el professor responsable d'imposar l'amonestació. CHAR(255).
- IdTipusamonestacio (Fk), relaciona cada amonestació imposada amb un tipus d'amonestació. Indica de quin tipus és l'amonestació. NUMBER (10,0).
- Data, aquest atribut indicarà quin dia ha estat imposada l'amonestació. DATE.
- Comunicatpares, aquest atribut indicarà si l'amonestació imposada a un alumne ha sigut comunicada als pares. Aquest camp serà S, si els pares han estat comunicats o, N si els pares encara no han estat comunicats. CHAR(1).

### 2.1.2.6 Entitat TIPUS SANCIO

Tipus sanció contindrà informació relativa a tots els tipus de sanció que es podran imposar als alumnes.

TIPUSS SANCIO és molt semblant a TIPUS AMONESTACIÓ, ens descriu els diferents tipus de sanció que pot rebre un alumne del centre. Com passava a TIPUS AMONESTACIÓ, amb aquesta taula es podrà crear qualsevol tipus de sanció d'una manera senzilla i àgil.

Com passa a la taula PROFESSOR, aquesta taula també conté un camp denominat ESTAT que guarda la informació de la presència o no d'un tipus de sanció al centre.

Tipus de sanció	Quan s'activa
Quedar-se una hora extra d'estudi durant una setmana	Automàticament quan l'alumne arriba tard tres vegades
Prova d'avaluació extra	Automàticament quan fan molt de soroll en classe 5 vegades
Sanció personalitzada	Introducció manual pel professor

Els atributs associats seran:

- IDTipussancio (Pk), descriu de forma única a un tipus de sanció. NUMBER(10,0).
- Descripció, descriu la sanció en un llenguatge humà.
- ReglaActivacio, descriu les regles que utilitzaran els professor per tal de imposar les sancions pertinents. CHAR(250).
- Vegades, aquest atribut indicarà els cops màxim que un alumne pot tenir acumulat d'amonestacions abans de que automàticament se l'imposi la sanció pertinent. NUMBER(10,0).

- Estat, aquest atribut ens indicarà si el tipus de sanció esta actiu, o sigui, si es pot posar una sanció per aquesta raó o no.

### 2.1.2.7 Entitat SANCIO

Sanció contindrà la informació relativa a totes les sancions que es posen en el centre.

SANCIO té una clau principal IDSANCIO per tal d'identificar únicament a una sanció introduïda. Els atributs IDALUMNE i IDTIPOUSSANCIO són claus foranes a les seves respectives taules, per tal d'indicar a una sanció qui la rep (IDALUMNE) i de que tipus (IDTIPOUSSANCIO).

L'atribut resolució podrà contenir un valor NULL fins que es doni una resolució. Per a aquesta tasca existeix el procediment que facilita al programador la tasca d'introducció d'una resolució.

Els atributs associats seran:

- IdSancio (Pk), identificador únic per identificar les sancions introduïdes. NUMBER(10,0).
- IdAlumne (Fk), identifica a l'alumne que ha rebut la sanció. CHAR(255).
- IdTipussancio (Fk), identifica al tipus de sanció que ha rebut l'alumne. NUMBER(10,0).
- DataCreació, defineix la data en que ha estat imposada la sanció. DATE.
- Resolució, indica la descripció de com ha anat la sanció. Podrà contenir un valor NULL fins que es doni una resolució. Per a aquesta tasca existeix un procediment que facilita al programador la tasca d'introducció d'un resolució. CHAR(250).

### 2.1.2.8 Entitat ASSIGNATURA

Assignatura contindrà informació relativa a totes les assignatures que s'imparteixen en el centre. ASSIGNATURA té dos atributs, un que fa la funció de clau principal i altre que defineix la descripció de l'assignatura.

Els requeriments no indiquen que s'hagi de emmagatzemar més informació, tot i que la BD\* estaria preparada per poder definir aquí nous atribut referents a l'assignatura tals com cost dels llibres, classe física on s'imparteix l'assignatura, etc.

Els atributs associats seran:

- IdAssignatura (Pk), identificador únic per identificar a les assignatures impartides en el centre.
- Descripció, defineix la descripció de l'assignatura.

**\*BD: Base de dades**

### 2.1.2.9 Entitat CALENDARI

CALENDARI contindrà la informació relativa als calendaris de cada un dels cursos que s'imparteixen al centre.

CALENDARI té una clau principal complexa composta per els 4 camps de la taula. Aquest fet és així perquè un calendari no podrà tenir dos assignatures a una mateixa hora el mateix dia de la setmana.

IDCALENDARI indicarà el nom del calendari i els camps IDASSIGNATURA, DIASETMANA i HORA indicaran dintre del calendari escolar que dia de la setmana s'imparteix una assignatura en concret.

Els atributs associats seran:

- IdCalendari (Pk), defineix un nou calendari. NUMBER(10,0).
- IdAssignatura (Pk, Fk), defineix quines assignatures estan assignades al calendari concret. NUMBER(10,0).
- DiaSetmana (Pk), ens indica els dies de la setmana que està associat a aquest horari. CHAR(255).
- Hora (Pk), ens indica a quina hora està associat aquest calendari. CHAR(255).

### 2.1.2.10 Entitat CURS

CURS contindrà informació relativa als cursos impartits en el centre educatiu. Està relacionat amb el calendari, ja que cada curs podrà disposar d'un calendari definit en CALENDARI.

CURS té una clau principal IDCURS. Té una clau forana Idprofessorresponsable que ens indica el professor que és responsable del curs.

Aquests cursos en un principi s'han establert en primer, segon, tercer i quart d'ESO però es deixa l'opció per a que en un futur el centre pugui oferir més cursos.

Els atributs associats seran:

- IdCurs (Pk), identifica de manera única a un curs impartit del centre. NUMBER(10,0).
- IdCalendari, identifica el calendari que té un curs en concret. NUMBER(10,0).
- IdProfessorResponsable (Fk), identifica al professor responsable d'un curs. CHAR(255).
- DataInici, indica quan donarà inici un curs. DATE.
- DataFi, indica quan donarà fi un curs. DATE.

### 2.1.2.11 Entitat MATRICULACIÓ

Matriculació contindrà informació relativa a totes les matriculacions que s'han fet en el centre en l'any en curs.

La clau primària d'aquesta taula és IDMATRICULACIO. Aquest atribut ens assegura que totes les matriculacions fetes són úniques. Ens identificarà la matrícula de cada alumne.

Disposa de dos atributs més que són claus foranes, l'identificador de l'alumne que es vol matricular i l'identificador del curs on vol matricular-se.

Aquesta entitat es podria posar més informació referent a la matrícula com pot ser, empleat administratiu que ha portat la matrícula, promocions adherides (com família nombrosa), data y hora de la matrícula...

No obstant el requeriments del treball es centra molt més en un sistema d'amonestacions i no pas en un gestor de matriculacions.

Els atributs associats seran:

- IdCurs (Fk), identifica al curs on es farà la matriculació. NUMBER(10,0).
- IdAlumne (Fk), identifica a l'alumne que s'està matriculant. CHAR(255).

### 2.1.2.12 Entitat DOCÈNCIA

DOCÈNCIA conté la relació que hi ha entre un professor, una assignatura i un curs. Emmagatzema la informació necessària per saber quin professor fa la classe d'una assignatura en concret en un curs donat.

La clau primària la formen els tres atributs definits. Aquests atributs són alhora les seves claus foranes respectivament de professor, curs i assignatura.

Per exemple, al curs 1 imaginem que està vinculat al calendari 2 que el dilluns a les 12h impartirà l'assignatura de física. Aquesta taula indica que l'assignatura física del curs 1 està impartida pel professor amb DNI (segons la taula professor).

DOCÈNCIA es necessària pel disseny per tal de poder donar un llistat de professors que donen les assignatures de cada curs independent del calendari al que esta vinculat el curs.

Els atributs associats seran:

- IdProfessor (Pk, Fk), defineix el professor. CHAR(255).
- IdCurs (Pk, Fk), defineix el curs. NUMBER(10,0).
- IdAssignatura (Pk, Fk), defineix l'assignatura. NUMBER(10,0).

### 2.1.2.13 Sistema de control

Per acabar es plantegen uns sistemes de control per tal de verificar el bon funcionament del projecte. Aquests seran:

- **Auditoria**, cada procediment tindrà agregat un insert a una taula auxiliar d'auditoria d'activitats on l'execució ens donarà el nom del procediment, els paràmetres d'entrada i sortida, la data i un identificador únic que ens identificarà de forma única els diferents canvis en la base de dades. L'auditoria serà anomenada LOG.
- **Resposta de control**, cada procediment contindrà un valor de sortida tipus CHAR que ens indicarà si el procediment que s'ha executat a la base de dades ha resultat correcte. Si es així ens donaria un 'OK', però si no ha resultat satisfactòria ens avisaria amb un error, tot indicant l'error que s'ha produït. La resposta de control serà anomenada RSP.

### 2.1.3 Diagrama del model entitat-relació

En el diagrama E/R queden reflectides totes les entitats, les relacions i els atributs necessaris per complir els requisits en la base de dades.

En el següent diagrama entitat relació es poden observar les entitats descrites anteriorment. Cada una d'aquestes estan relacionades amb altres, fent així un conjunt força compacte. Es començarà explicant com al llarg d'aquestes 15 setmanes han anat evolucionant cadascuna d'aquestes entitats.

En un primer moment es va fer servir una entitat anomenada PERSONA que va englobar les característiques comunes de les entitats PROFESSOR i ALUMNE, tenint aquestes últimes només atributs propis, com per exemple en PROFESSOR horariAtencioPares.

Quan s'anava avançant en la construcció dels mètodes es va veure que aquesta idea no era massa bona ja que es veien problemes en fer alguns procediments requerits en l'enunciat. Es pot veure que en aquest diagrama final no surt aquesta entitat PERSONA.

Una altra entitat que es va eliminar del primer esbós va ser l'entitat PAREMAREALUMNE, que en un principi funcionava com detector de les dades dels pares de cada alumne i al final es va veure que no calia tenir-lo en una taula diferent.

En total hi ha 12 entitats diferents totes elles relacionades, les altres taules no relacionades del diagrama han calgut per a resoldre el mòdul estadístic i alguns altres requisits com és el cas de la taula LOG.

Es començarà explicant les relacions que hi ha entre les diferents entitats:

- Relació ALUMNE-TUTOR-PROFESSOR on cada alumne tindrà un únic tutor, però el mateix tutor podrà tenir a més d'un alumne per tutoritzar. A més el professor podrà o no ser tutor i podrà tenir tutoritzats a més d'un alumne a la vegada.

Un tutor no és podrà donar d'alta si el professor o l'alumne no existeix en les taules corresponents (claus foranes en TUTOR, IdAlumne, IdProfessor).

- Relació ALUMNE-MATRÍCULA on cada alumne podrà tenir associada una matrícula o no tenir-ne cap. Aquesta matrícula no és podrà donar d'alta si l'alumne en qüestió no existeix en la taula ALUMNE (clau forana en MATRICULA, IdAlumne).
- Relació ALUMNE-AMONESTACIÓ-PROFESSOR on cada amonestació serà imposada per un únic professor a un únic alumne.  
L'alumne port rebre moltes amonestacions o no rebre cap i un professor podria posar moltes amonestacions o no posar-ne cap. Una amonestació no podrà ser donada d'alta si no existeix l'alumne o no existeix el professor en les seves respectives taules (claus foranes en AMONESTACIÓ IdAlumne, IdProfessor).
- Relació TIPUSAMONESTACIÓ-AMONESTACIÓ en l'entitat TIPUSAMONESTACIO només s'indica els tipus d'amonestació que el centre té definits.  
Una amonestació només podrà ser d'un sol tipus, però del mateix tipus podem haver-hi més d'una amonestació imposada.  
Una amonestació no es podrà donar d'alta si no existeix el tipus d'amonestació (clau forana a AMONESTACIÓ IdTipusamonestacio).
- Relació TIPUSSANCIÓ-SANCIÓ, en l'entitat TIPUSSANCIÓ només s'indica el tipus de sanció que podrà ser imposat per un professor a un alumne o bé per una acumulació d'amonestacions.  
Així una sanció només podrà ser d'un sols tipus, però del mateix tipus podem haver-hi més d'una sanció imposada.  
Una sanció no es podrà donar d'alta si no existeix el tipus de sanció (clau forana a SANCIÓ IdTipussancio).
- Relació SANCIÓ-AMONESTACIÓ aquesta relació esta present perquè les sancions als alumnes podran imposar-se automàticament quan un alumne tingui acumulació d'amonestacions.
- Relació MATRICULACIÓ-CURS on un curs pot tenir moltes matriculacions i una matriculació només pot estar vinculada a un curs en concret.  
Una matricula no es podrà donar d'alta si no existeix el curs en concret en la taula CURS (clau forana a MATRICULACIÓ IdCurs).
- Relacions CURS-DOCÈNCIA, ASSIGNATURA-DOCÈNCIA i PROFESSOR-DOCÈNCIA aquestes tres relacions les explicarem juntes perquè la taula docència engloba les tres. Docència serà la impartició d'una classe i una

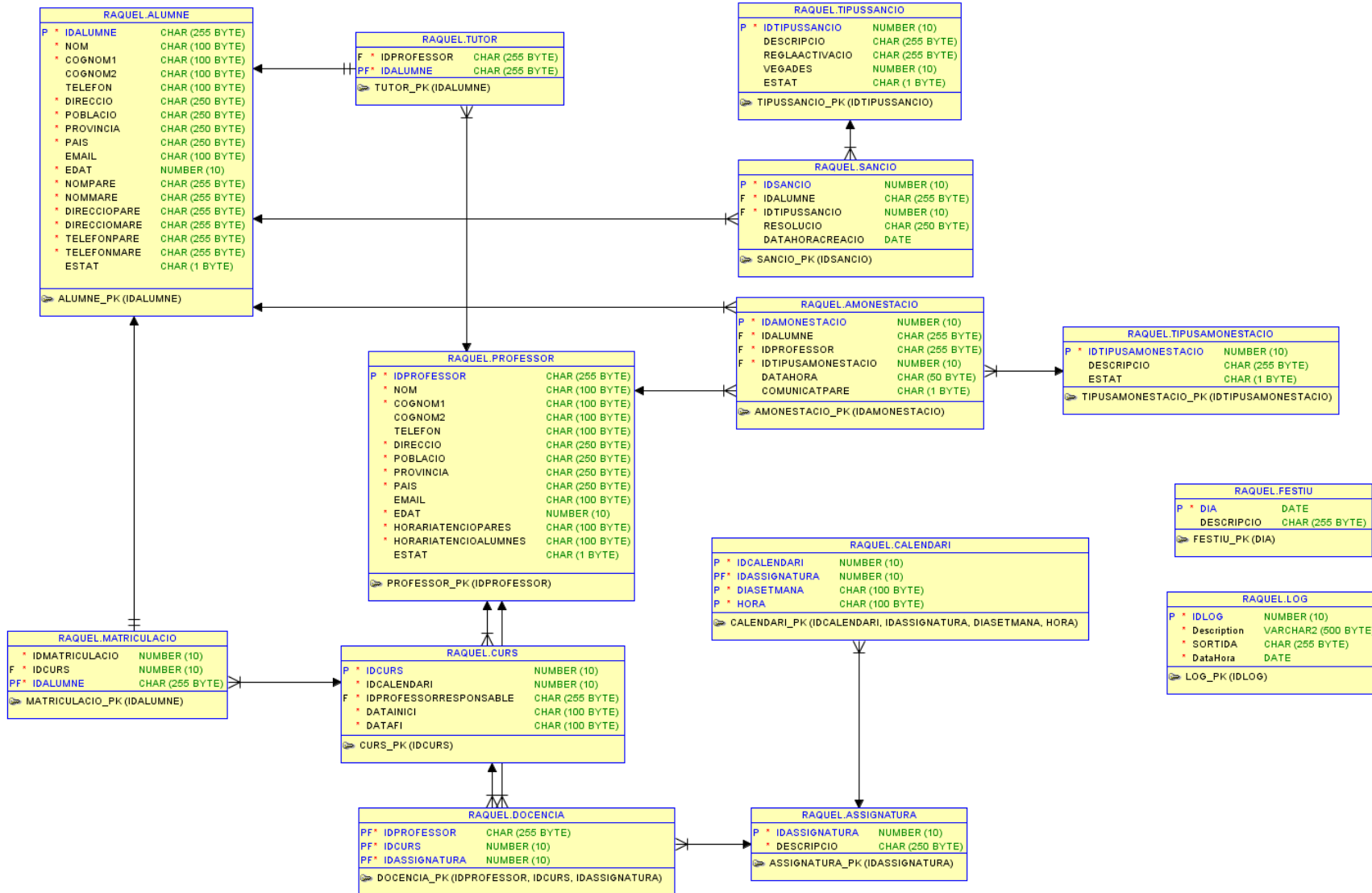
classe és sempre en un curs determinat amb una assignatura en concret i amb el professor que imparteix aquesta assignatura, per tant la clau primària de DOCÈNCIA és el conjunt dels seus atributs que a la vegada són claus foranes de les seves respectives taules IdAssignatura, IdProfessor i IdCurs.

- Relació ASSIGNATURA-CALENDARI on cada assignatura tindrà el seu calendari corresponent determinat pel dia de la setmana i l'hora d'impartició.

Un calendari no és podrà donar d'alta sense tenir en la taula ASSIGNATURA l'assignatura en concret (clau forana a CALENDARI IdAssignatura).

- Relació PROFESSOR-CURS on cada curs tindrà assignat un professor responsable.

No és podrà donar d'alta un curs si el professor en qüestió no hi és a la taula PROFESSOR (clau forana a CURS IdProfessorresponsable).





### **3. Disseny físic del projecte**

#### **3.1 Introducció: Transformació al model relacional**

Les entitats abans esmentades formen part del disseny conceptual, originen les relacions i les interrelacions. Aquestes relacions es transformen com es veu en l'apartat 2 en claus foranes d'alguna relació ja establerta o en noves relacions.

El funcionament del gestor serà independent, imaginem que només hi ha un únic centre educatiu (encara que la base de dades podria ser adaptada fàcilment per a que pugui formar part d'un sistema distribuït amb una única base de dades identificant la informació en funció del centre).

Aquest projecte només contindrà els alumnes i professor d'un centre on només podran accedir a les dades les persones autoritzades a aquesta tasca, com poden ser les administratives del centre o els diferents professors.

En aquest punt s'exposa els diferents procediments emmagatzemats que faran interactuar al programador amb la base de dades. El programador confiarà en que el disseny exposat es correcte i només es preocuparà de fer servir correctament els procediments documentats.

Es per això que totes les operacions que es poden fer a la BD hauran de ser realitzades per un dels procediments que es presenten en aquest apartat amb la seva corresponent documentació d'ús.

La seva utilització es senzilla, només cal nombrar el SP al que volem cridar, i afegir-hi els paràmetres que necessita. El SP en tots els casos retornarà una paràmetre de sortida on serà "OK" si l'execució d'aquest ha sigut correcte o bé "ERROR + Tipus d'error" en cas de que hi hagi fallat quelcom durant la seva execució. Aquest paràmetre de sortida és un dels requeriments indicats.

Primer es defineixen els SP per la creació de les taules, a continuació es defineixen els SP d'alta, baixa i modificació de les diferents entitats de la BD, com alta d'un alumne, un tipus de sanció o una matriculació...

Per últim es procedirà a la informació necessària de les llistes requerides i al joc de proves per tal de confirmar el bon funcionament de la BD.

## **3.2 Creació dels scripts i procedures**

### **3.2.1 Creació de les taules**

Per a començar a formar la BD es crearà primer les taules de les anteriors entitats que ens faran falta. Alhora de crear les taules només s'utilitzarà tres tipus diferents de dades CHAR, NUMBER i DATE, aquests tipus permeten guardar totes les dades necessàries i a més, són fàcil de fer servir.

#### **3.2.1.1 Taula PROFESSOR**

Aquesta taula registra els diferents professors que treballen en el centre, tot indicant les dades personals (atributs comentats en l'apartat anterior).

Per tal de poder crear un nou professor, aquest no pot existir amb anterioritat a la taula. Això significa que la taula tindrà una clau primària que serà IDPROFESSOR que ens indicarà si el professor està ja inserit o no.

#### **3.2.1.2 Taula ALUMNE**

Aquesta taula registra els diferents alumnes que estaran matriculats en el centre, tot indicant les dades personals (atributs comentats en l'apartat anterior).

Per tal de poder crear un nou alumne, aquest no pot existir amb anterioritat a la taula. Això significa que la taula tindrà una clau primària que serà IDALUMNE que ens indicarà si l'alumne està ja inserit o no.

#### **3.2.1.3 Taula TUTOR**

Aquesta taula registra els diferents tutors que pot tenir un alumne, tot indicant el IDALUMNE i el IDPROFESSOR (atributs comentats en l'apartat anterior).

Per tal de poder crear un nou tutor, s'haurà de complir que IDALUMNE i el IDPROFESSOR existeixen en les seves respectives taules. La taula conté dos claus foranes a aquests dos atributs i per tal de verificar que l'alumne tingui només un tutor té com a clau primària IDALUMNE.

#### **3.2.1.4 Taula TIPUAMONESTACIO**

Aquesta taula registra els diferents tipus d'amonestació que es poden imposar en el centre tot indicant les característiques de l'amonestació (atributs comentats en l'apartat anterior).

Per tal de poder crear un nou tipus d'amonestació, aquest tipus no pot existir amb anterioritat a la taula. Això significa que la taula tindrà una clau primària que serà IDTIPUSAMONESTACIO que ens indicarà si aquest tipus està ja inserit o no.

#### **3.2.1.5 Taula AMONESTACIO**

Aquesta taula registra les diferents amonestacions imposades en el centre tot indicant l'alumne que ha rebut l'amonestació, el professor que ha imposat l'amonestació i el tipus d'amonestació.

Per tal de poder crear una nova amonestació, aquesta no pot existir amb anterioritat a la taula. Això significa que la taula tindrà una clau primària que serà IDAMONESTACIO que ens indicarà si aquesta amonestació està ja inserida o no.

S'ha de tenir en compte que l'alumne amonestat, el professor amonestador i el tipus d'amonestació imposat tenen que existir en les seves respectives taules. És així perquè AMONESTACIO té tres claus foranes en aquests tres atributs.

#### **3.2.1.6 Taula TIPUSSANCIO**

Aquesta taula registra els diferents tipus de sancions que es poden imposar en el centre tot indicant les característiques de la sanció (atributs comentats en l'apartat anterior).

Per tal de poder crear un nou tipus de sanció, aquesta no pot existir amb anterioritat a la taula. Això significa que la taula tindrà una clau primària que serà IDTIPUSSANCIO que ens indicarà si aquest tipus està ja inserit o no.

#### **3.2.1.7 Taula SANCIO**

Aquesta taula registra les diferents sancions imposades en el centre tot indicant l'alumne que ha rebut la sanció, el tipus de sanció, la resolució i la data de creació.

Per tal de poder crear una nova sanció, aquesta no pot existir amb anterioritat a la taula. Això significa que la taula tindrà una clau primària que serà IDSANCIO que ens indicarà si aquesta està ja inserida o no.

S'ha de tenir en compte que l'alumne sancionat i el tipus de sanció imposat tenen que existir en les seves respectives taules per tal de poder inserir la sanció. És així perquè SANCIO té dos claus foranes en aquests dos atributs.

### **3.2.1.8 Taula ASIGNATURA**

Aquesta taula registra les diferents assignatures que s'imparteixen en un curs.

Per tal de poder crear un nou tipus d'assignatura aquesta no pot existir amb anterioritat en la taula. Això significa que la taula tindrà una clau primària que serà IDASSIGNATURA que ens indicarà si aquesta està ja inserida o no.

### **3.2.1.9 Taula CALENDARI**

Aquesta taula registra els diferents calendaris que té cada assignatura i per tant, cada curs.

Per tal de poder crear un nou calendari aquest no pot existir amb anterioritat a la taula. Això significa que la taula tindrà una clau primària que serà IDCALENDARI que ens indicarà si aquest està inserit o no amb anterioritat.

CALENDARI també conté una clau forana a ASSIGNATURA que indica que no es podrà donar d'alta un calendari si l'assignatura en que es regeix aquest calendari no existeix. Primer s'haurà de crear l'assignatura i després el seu calendari.

### **3.2.1.10 Taula CURS**

Aquesta taula registra els diferents cursos que imparteix el centre escolar, indiquen els cursos que els alumnes es poden matricular.

Per tal de poder crear un nou curs aquest no pot existir ja en la taula. Això significa que la taula tindrà una clau primària que serà IDCURS que indicarà si aquest està inserit o no amb anterioritat.

CURS també conté una clau forana a PROFESSOR que ens indica que no podrem donar d'alta un curs si abans no estan donat d'alta el professor.

El calendari pel qual es regeix el curs també a d'estar inserit amb anterioritat a la taula CALENDARI.

### **3.2.1.11 Taula MATRICULACIÓ**

Aquesta taula registra les diferents matriculacions dels alumnes del centre. Aquestes matriculacions indiquen el curs on els alumnes s'han matriculat.

Per tal de poder crear una nova matriculació aquesta no pot existir amb anterioritat en la taula. Això significa que la taula tindrà una clau primària que serà IDMATRICULACIO que ens indicarà si aquesta està inserida o no amb anterioritat.

MATRICULACIÓ també conté clau forana a ALUMNE i CURS que ens indica que no es podrà donar d'alta una matriculació si abans no estan donats d'alta l'alumne i el curs en les seves taules respectives.

### **3.2.1.12 Taula DOCÈNCIA**

Aquesta taula registra les diferents docències del centre. Docència ens relaciona el professor amb el curs i amb l'assignatura que imparteix.

Per tal de poder crear una nova docència tindrà que existir el professor, el curs i l'assignatura a la qual és refereix. Aquests tres atributs són claus foranes de docència i no pot existir una docència si no estan donats d'alta amb anterioritat aquest atributs.

### **3.2.1.13 Taula FESTIU**

Aquesta taula ha estat creada per donar resposta al requisit de conèixer els dies festius que dona la conselleria d'educació en un any lectiu.

Aquesta taula conté els següents atributs:

- Dia, és la clau primària. No pot haver-hi dos dies idèntics en la taula.
- Descripció, descriurà el tipus de festa que és.

### **3.2.1.14 Taula LOG**

Per tal de poder auditar tota l'activitat de la BD feta pels procediments, cal una taula on guardar la informació de les modificacions fetes a la base de dades.

Així doncs es podrà saber quin dia s'ha executat una operació concreta.

Aquesta taula serà cridada per tots els procediments quan es produeixi un accés a la BD i així registrar l'acció realitzada inserint les dades.

Cada inserció tindrà una clau única, això es gràcies a la clau primària de la taula LOG.

### **3.2.1.15 Taules estadístiques**

Per tal de poder donar a temps real les estadístiques, calen unes taules auxiliars que serveixen d'emmagatzematge de totes les estadístiques actualitzades del total de l'activitat de la BD.

Aquestes taules donaran resposta als requeriments estadístics. Per veure aquestes dades només ens caldrà fer un SELECT a alguna de les taules per a respondre a les qüestions estadístiques.

En total tenim vuit qüestions estadístiques que s'exposaran ara.

- 1- El número d'amonestacions per alumne (independentment del curs), per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. En aquesta taula per tant, s'emmagatzemaran el número d'amonestacions que té un alumne.

```
--CONSULTA NÚMERO 1
CREATE TABLE "E_AMONESTACIO_ALUMNE"
(
  "IDALUMNE" CHAR(255),
  "NUM_AMONESTACIONS" NUMBER,
CONSTRAINT "E_AMONESTACIO_ALUMNE_PK" PRIMARY KEY ("IDALUMNE")
ENABLE
)
```

- 2- El número de sancions per alumne i any. per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. En aquesta taula per tant, s'emmagatzemaran el número de sancions que té un alumne en l'any en curs.

```
--CONSULTA NÚMERO 2
CREATE TABLE "E_SANCIO_ALUMNE"
(
  "IDALUMNE" CHAR(255),
  "NUM_SANCIONS" NUMBER,
  "ANY" CHAR(255),
CONSTRAINT "E_SANCIO_ALUMNE_PK" PRIMARY KEY ("IDALUMNE") ENABLE
)
```

- 3- La mitjana d'amonestacions per professor i any, per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. En aquesta taula per tant, s'emmagatzemaran la mitjana d'amonestacions per professor i en l'any en curs.

```
--CONSULTA NÚMERO 3
CREATE TABLE "E_MITJANA_PROFESSOR"
(
  "IDPROFESSOR" CHAR(255) NOT NULL ENABLE,
  "MITJANA" NUMBER,
  "ANY" CHAR(255),
  "NOM" CHAR(255),
CONSTRAINT "E_MITJANA_PROFESSOR_PK" PRIMARY KEY
("IDPROFESSOR") ENABLE
)
```

- 4- El número de sancions per curs i any, per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. En aquesta taula per tant, s'emmagatzemaran el número de sancions que hi ha en un curs en l'any vigent.

```
--CONSULTA NÚMERO 4
CREATE TABLE "E_SANCIONS_CURS"
(
  "IDCURS" NUMBER,
  "NUM_SANCIONS" NUMBER,
  "ANY" CHAR(255)
)
```

- 5- El nom de l'alumne més sancionat en un any donat, per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. Aquesta emmagatzemarà el nom de l'alumne que en un moment determinat tingui més sancions.

```
--CONSULTA NÚMERO 5
CREATE TABLE "E_ALUMNE_MES_SANCIONAT"
(
  "NOM" CHAR(255),
  "ANY" CHAR(255),
  CONSTRAINT "E_ALUMNE_MES_SANCIONAT_PK"
  PRIMARY KEY ("NOM") ENABLE
)
```

- 6- El nom del professor més amonestador per curs, per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. En aquesta taula per tant, s'emmagatzemaran el nom del professor amb més amonestacions creades en determinat curs.

```
--CONSULTA NÚMERO 6
CREATE TABLE "E_PROFESSOR_MES_AMONESTADOR"
(
  "IDPROFESSOR" CHAR(255) NOT NULL
  ENABLE,
  "NOM" CHAR(255) NOT NULL ENABLE,
  "IDCURS" NUMBER,
  CONSTRAINT "MESAMONESTADOR_PK"
  PRIMARY KEY ("IDPROFESSOR")
  ENABLE
)
```

- 7- La mitjana de sancions que tenen els alumnes per curs, per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. Aquesta emmagatzemarà la mitjana de sancions que tenen els alumnes en un curs.

```
--CONSULTA NÚMERO 7
CREATE TABLE "E_MITJANA_SANCIONS_ALUMNE"
(
  "IDALUMNE" CHAR(255) NOT NULL
  ENABLE,
  "MITJANA" NUMBER,
  "NOM" CHAR(255),
  "CURS" NUMBER,
  CONSTRAINT "E_MITJANA_SANCIONS_ALUMNE_PK"
  PRIMARY KEY ("IDALUMNE") ENABLE
)
```

- 8- El número d'alumnes que no tenen cap amonestació, per tal de donar resposta a aquesta qüestió s'ha creat la següent taula. Aquesta emmagatzemarà el número d'alumnes que no tenen cap amonestació.

```
--CONSULTA NÚMERO 8
CREATE TABLE "E_CAP_AMONESTACIO"
(
  "NUMALUMNES" NUMBER,
  CONSTRAINT "E_CAP_AMONESTACIO_PK"
  PRIMARY KEY ("NUMALUMNES") ENABLE
)
```

### 3.2.2 Creació dels procediments d'alta, de baixa i de modificació

Un cop creades les taules es pot començar a crear els procediments per inserir, esborrar i modificar les dades de les anteriors taules descrites.

Un cop creades les taules, per a cadascuna d'elles, hi haurà tres tipus de procediment:

- Procediment d'alta.
- Procediment de baixa.
- Procediment de modificació.

Es començarà a descriure els diferents procediments que s'utilitzaran pel maneig de base de dades. Tots els procediments tindran paràmetres d'entrada i de sortida de tipus CHAR, NUMBER o DATE.

#### 3.2.2.1 PROFESSOR

Els tres tipus de procediments alta, baixa i modificació seran anomenats respectivament:

- SP\_ALTA\_PROFESSOR, aquest procediment permet donar d'alta a un professor del centre.  
Els seus paràmetres d'entrada són tots els atributs de professor (Idprofessor, nom, cognom1, cognom2, telèfon, direcció, població, província, país, email, edat, horari atenció als pares, horari d'atenció als alumnes i estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos professors no poden tenir dos Idprofessor iguals. Es tindrà en compte que no es podrà ficar un DNI que ja consti com un alumne.
- SP\_BAIXAPROFESSOR, aquest procediment permet donar de baixa a un professor. S'ha de tenir en compte l'històric dels professors que han passat pel centre i que , per tant, quan un professor marxa del centre no volem que s'esborri de la base de dades sinó que figuri com no actiu.  
Per tant, el procediment de baixa s'utilitzarà quan s'introdueixi un professor que al final no acaba mai donant classe (exemple: un professor ha sigut assignat a un centre per tal de que comenci les seves classes el dia 1 d'abril, en el centre li donen d'alta abans de que arribi però el professor no arriba el dia 1 d'abril perquè a sigut assignat en últim moment a un altre centre).  
Tindrà un paràmetre d'entrada, l'atribut Idprofessor.



Tindrà un paràmetre de sortida RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

El procediment llançaria un error si l'Idprofessor posat com a paràmetre d'entrada no existís, perquè no es podrà esborrar a un professor que no existeix.

- SP\_MOD\_PROFESSOR, aquest procediment permet modificar qualsevol valor dels atributs d'un professor com per exemple la seva adreça (en el cas que un professor canviï de direcció).

Quan un professor que ha donat classe en el centre es trasllada a un altre centre o bé es jubila, es farà servir aquest procediment per tal de posar l'atribut ESTAT de la taula professor a un estat 'N', no actiu.

Amb aquest procediment, per tant, es podrà tenir un històric de tots els professors que han passat pel centre.

Els paràmetres d'entrada seran tots els atributs de professor (Idprofessor, nom, cognom1, cognom2, telèfon, direcció, població, província, país, email, edat, horari atenció als pares, horari d'atenció als alumnes i estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

El procediment llançaria un error si l'Idprofessor posat com a paràmetre d'entrada no existís, perquè no es podrà modificar a un professor que no existeix.

### 3.2.2.2 ALUMNE

Els tres tipus de procediments alta, baixa i modificació seran anomenats respectivament:

- SP\_ALTA\_ALUMNE, aquest procediment permet donar d'alta als alumnes del centre.

Els seus paràmetres d'entrada són els atributs especificats en la taula alumne (Idalumne, nom, cognom1, cognom2, telèfon, direcció, població, província, país, email, edat, nom pare, nom mare, direcció pare, direcció mare, telèfon pare, telèfon mare, estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos alumnes no poden tenir dos Idalumne iguals. Es tindrà en compte que no es podrà ficar un DNI que ja consti com un professor.

- SP\_BAIXAALUMNE, aquest procediment permet donar de baixa a un alumne. S'ha de tenir en compte l'històric dels alumnes que han passat pel centre i que , com en el cas dels professors, quan un alumne marxa del centre no s'esborra de la base de dades sinó que figurarà com no actiu. Per tant, el procediment de baixa s'utilitzarà quan s'introdueixi un alumne que al final no acaba mai matriculat en el centre (exemple: un alumne ha sigut assignat a un centre per tal de que comenci en el curs primer d'ESO l'any vinent, en el centre li donen d'alta abans de que comenci el curs però l'alumne no arriba mai a matricular-se en el centre). Tindrà un paràmetre d'entrada, l'atribut Idalumne. Tindrà un paràmetre de sortida RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error. El procediment llançaria un error si l'Idalumne posat com a paràmetre d'entrada no existís, perquè no es podrà esborrar a un alumne que no existeix.
  
- SP\_MOD\_ALUMNE, aquest procediment permet modificar qualsevol valor dels seus atributs de l'alumne com per exemple la seva adreça (en el cas que un alumne canviï de direcció). Quan un alumne que ha estat matriculat en el centre es trasllada a un altre centre o bé es gradua, es farà servir aquest procediment per tal de posar l'atribut ESTAT de la taula alumne a un estat 'N', no actiu. Amb aquest procediment, per tant, es podrà tenir un històric de tots els alumnes que han passat pel centre. Els paràmetres d'entrada seran tots els atributs d'alumne (Idalumne, nom, cognom1, cognom2, telèfon, direcció, població, província, país, email, edat, nom pare, nom mare, direcció pare, direcció mare, telèfon pare, telèfon mare, estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error. El procediment llançaria un error si l'Idalumne posat com a paràmetre d'entrada no existís, perquè no es podrà modificar a un alumne que no existeix.

### 3.2.2.3 CURS

Els tres tipus de procediments alta, baixa i modificació seran anomenats respectivament:

- SP\_ALTA\_CURS, aquest procediment permet donar d'alta els diferents cursos que imparteix el centre. Els seus paràmetres d'entrada són els atributs especificats en la taula curs (Idcurs, Idcalendari, Idprofessorresponsable, data inici curs i data fi curs). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i

donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos cursos no poden tenir dos Idcurs iguals. S'ha de tenir en compte també la seva clau forana Idprofessorresponsable, si no existeix un professor amb Idprofessor = Idprofessorresponsable també donaria error.

Passaria el mateix, donarà error, si en la taula CALENDARI no existeix un calendari.Idaclendari = curs.Idcalendari.

No es podrà inserir un curs si el professor amb Idprofessorresponsable té com a ESTAT='N'.

- SP\_BAIXACURS, aquest procediment permet donar de baixa un curs. Tindrà un paràmetre d'entrada, l'atribut Idcurs. Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error. El procediment llançaria un error si l'Idcurs posat com a paràmetre d'entrada no existís, perquè no es podrà esborrar un curs que no existeix. S'ha de tenir en compte que abans d'esborrar un curs s'ha d'esborrar primer les matricules i les docències que s'han fet en el centre (perquè matriculació i docència contenen Idcurs com una clau forana).
- SP\_MOD\_CURS, aquest procediment permet modificar un curs existent. Els paràmetres d'entrada seran tots els atributs de curs (Idcurs, Idcalendari, Idprofessorresponsable, data inici curs i data fi curs). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error. El procediment llançaria un error si l'Idcurs posat com a paràmetre d'entrada no existís, perquè no es podrà modificar un curs que no existeix. També llançaria un error si el Idprofessorresponsable posat com a paràmetre o Idcalendari no existissin en les seves taules corresponents PROFESSOR i CALENDARI.

#### 3.2.2.4 ASSIGNATURA

Els tres tipus de procediments alta, baixa i modificació seran anomenats respectivament:

- SP\_ALTA\_ASSIGNATURA, aquest procediment permet donar d'alta les diferents assignatures que imparteixen els cursos al centre. Els seus paràmetres d'entrada són els atributs especificats en la taula assignatura (Idassignatura i descripció). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos assignatures no poden tenir dos Idassignatura iguals.

- SP\_BAIXAASSIGNATURA, aquest procediment permet donar de baixa una assignatura.  
Tindrà un paràmetre d'entrada, l'atribut Idassignatura. Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
El procediment llançaria un error si l'Idassignatura posat com a paràmetre d'entrada no existís, perquè no es podrà esborrar una assignatura que no existeix.  
S'ha de tenir en compte que abans d'esborrar una assignatura s'ha d'esborrar primer el seu calendari (perquè calendari conté Idassignatura com una clau forana).
- SP\_MOD\_ASSIGNATURA, aquest procediment permet modificar una assignatura existent.  
Els paràmetres d'entrada seran tots els atributs de l'assignatura (Idassignatura i descripció). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
El procediment llançaria un error si l'Idassignatura posat com a paràmetre d'entrada no existís, perquè no es podrà modificar una assignatura que no existeix.

### 3.2.2.5 MATRICULACIÓ

En aquesta taula només es permetrà la inserció, o sigui l'alta de la matrícula. No es permetrà en cap moment la modificació ni la baixa un cop s'hagi donat d'alta la matrícula d'un alumne. Matriculació només tindrà un procediment el de donar d'alta les matriculacions anomenat SP\_ALTA\_MATRICULACIO.

Aquest procediment com s'ha comentat permet donar d'alta les diferents matrícules dels alumnes. Els seus paràmetres d'entrada són els atributs especificats en la taula matriculació (Idmatriculacio, Idcurs, Idalumne). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos matrícules no poden tenir dos Idalumne iguals ja que només esta permès una matricula per alumne.

S'ha de tenir en compte també la seves claus foranes Idalumne i Idcurs, en aquest cas si no existeix un alumne amb alumne.Idalumne = matriculacio.Idalumne o no existeix un curs amb curs.Idcurs = matriculacio.Idcurs també donaria error.

### 3.2.2.6 CALENDARI

Els tres tipus de procediments alta, baixa i modificació seran anomenats respectivament:

- SP\_ALTA\_CALENDARI, aquest procediment permet donar d'alta els diferents calendaris que tindran les assignatures dels cursos impartits en el centre.

Els seus paràmetres d'entrada són els atributs especificats en la taula calendari (Idcalendari, Idassignatura, dia setmana i hora). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos calendaris no poden tenir dos (Idcalendari, Idassignatura, dia setmana, hora) iguals.

S'ha de tenir en compte també la seva clau forana Idassignatura, en aquest cas si no existeix una assignatura amb Assignatura.Idassignatura = Calendari.Idassignatura també donaria error, ja que un calendari té que estar vinculat sempre a una assignatura. Això és així perquè s'ha considerat Idcalendari com el nom del calendari on es fica tot l'horari escolar. D'aquesta manera es podrà concretar un horari vinculat a un calendari. Com exemple es tindria que el calendari pot tenir l'assignatura 1 el dilluns a les 8 del matí, el calendari 1 té també la assignatura 2 a les 9 del matí...

- SP\_BAIXACALENDARI, aquest procediment permet donar de baixa un calendari.

Tindrà un paràmetre d'entrada, l'atribut Idcalendari. Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

El procediment llançaria un error si l'Idcalendari posat com a paràmetre d'entrada no existís, perquè no es podrà esborrar un calendari que no existeix.

S'ha de tenir en compte que abans d'esborrar un calendari que s'ha d'esborrar primer el curs que té associat aquest calendari (perquè curs té associat Idcalendari).

- SP\_MOD\_CALENDARI, aquest procediment permet modificar un calendari existent.

Els paràmetres d'entrada seran tots els atributs del calendari (Idcalendari, Idassignatura, dia setmana i hora). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

El procediment llançaria un error si l'Idcalendari posat com a paràmetre d'entrada no existís, perquè no es podrà modificar un calendari que no existeix. També posaria un error si Idassignatura posat com a paràmetre d'entrada no existís.

### 3.2.2.7 TIPUSAMONESTACIO

Els tres tipus de procediments alta, baixa i modificació seran anomenats respectivament:

- SP\_ALTA\_TIPUS\_AMONESTACIO, aquest procediment permet donar d'alta els tipus d'amonestació que el centre cregui oportuns poder imposar als alumnes per un mal comportament.  
Els seus paràmetres d'entrada són els atributs especificats en la taula tipus amonestació (Idtipusamonestacio, descripció i estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos tipus d'amonestació no poden tenir dos Idtipusamonestacio iguals.
- SP\_BAIXA\_TIPUS\_AMONESTACIO, aquest procediment permet donar de baixa a un tipus d'amonestació. S'ha de tenir en compte l'històric dels tipus d'amonestacions que han passat pel centre i que, com en el cas dels professors, quan un tipus d'amonestació ja no es pot imposar con un mal comportament de l'alumne no s'esborra de la base de dades sinó que figurarà com no actiu.  
Per tant, el procediment de baixa s'utilitzarà quan s'introdueixi un tipus d'amonestacions que al final no acaba mai d'imposar-se en el centre (exemple: s'ha pensat de crear un nou tipus d'amonestacions quan comenci el curs escolar, en el centre li donen d'alta abans de que comenci el curs però el tipus d'amonestació no acaba d'agradar i, per tant, els professor obtén per no imposar mai aquest tipus).  
Tindrà un paràmetre d'entrada, l'atribut Idtipusamonestacio.  
Tindrà un paràmetre de sortida RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
El procediment llançaria un error si l'Idtipusamonestacio posat com a paràmetre d'entrada no existís, perquè no es podrà esborrar a cap tipus d'amonestació si no existeix.

- SP\_MOD\_TIPUS\_AMONESTACIO, aquest procediment permet modificar qualsevol valor dels atributs de tipus amonestació com per exemple la seva descripció.

Quan un tipus d'amonestació no agradi als docents dels centres perquè, per exemple, cap alumne comet aquest tipus d'amonestació, es farà servir aquest procediment per tal de posar l'atribut ESTAT de la taula tipus amonestació a un estat 'N', no actiu.

Amb aquest procediment, per tant, es podrà tenir un històric de tots els tipus d'amonestacions que han passat pel centre.

Els paràmetres d'entrada seran tots els atributs de tipus amonestació (ldtipusamonestació, descripció i estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

El procediment llançaria un error si l'ldtipusamonestacio posat com a paràmetre d'entrada no existís, perquè no es podrà modificar el tipus d'amonestació si no existeix.

### 3.2.2.8 TIPUSSANCIO

Els tres tipus de procediments alta, baixa i modificació seran anomenats respectivament:

- SP\_ALTA\_TIPUS\_SANCIO, aquest procediment permet donar d'alta als diferents tipus de sanció del centre.

Els seus paràmetres d'entrada són els atributs especificats en la taula tipus sanció (ldtipussancio, descripció, regla activació, vegades i estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos tipus de sanció no poden tenir dos ldtipussancio iguals.

Aquest procediment és especial. Al donar d'alta un tipus de sanció, cal indicar al camp REGLAACTIVACIO, la DESCRIPCIO del tipus de sanció i emplenar el camp VEGADES amb un nombre.

Sempre que les amonestacions amb la descripció indicada a la reglaactivacio superin el camp vegades es crearà una sanció del tipus indicat a l'atribut DESCRIPCIO. Per exemple:

Com diu els requeriments s'insereixi una sanció automàtica quan un alumne tingui 3 amonestacions per arribar tard.

```
SP_ALTA_TIPUS_SANCIO
      (IDTIPOUSSANCIO IN NUMBER,
      DESCRIPCIO IN CHAR,
      REGLAACTIVACIO IN CHAR,
      VEGADES IN NUMBER,
      RSP OUT CHAR);
```

Les taules TIPUSAMONESTACIO i TIPUSSANCIO quedarien d'aquesta manera:

IDTIPUSAMONESTACIO	DESCRIPCIO
1	alumne_arriba_tard
2	alumne_soroll
3	alumne_malparlat
4	alumne_no_deures

IDTIPOUSSANCIO	DESCRIPCIO	REGLAACTIVACIO	VEGADES
1	Hora extra estudi	alumne_arriba_tard	3
2	Avaluacio extra	alumne_soroll	5

Com es pot veure el camp Tipussancio.Reglaactivacio és OBLIGATORI que coincideixi amb el camp Tipusamonestacio.Descripcio perquè puguin funcionar les alarmes automàtiques depenen del camp VEGADES.

- SP\_BAIXA\_TIPUS\_SANCIO, aquest procediment permet donar de baixa a un tipus de sanció. S'ha de tenir en compte l'històric dels tipus de sancions que han passat pel centre i que, com en el cas dels tipus d'amonestació, quan un tipus de sanció ja no es pot imposar con un mal comportament de l'alumne no s'esborra de la base de dades sinó que figurarà com no actiu. Per tant, el procediment de baixa s'utilitzarà quan s'introdueixi un tipus de sanció que al final no acaba mai d'imposar-se en el centre (exemple: s'ha pensat de crear un nou tipus de sanció quan comenci el curs escolar, en el centre li donen d'alta abans de que comenci el curs però aquest tipus de sanció no acaba d'agradar i, per tant, els professor obtén per no imposar mai aquest tipus).  
Tindrà un paràmetre d'entrada, l'atribut Idtipussancio.  
Tindrà un paràmetre de sortida RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
El procediment llançaria un error si l'Idtipussancio posat com a paràmetre d'entrada no existís, perquè no es podrà esborrar a cap tipus de sanció si no existeix.



- SP\_MOD\_TIPUS\_SANCIO, aquest procediment permet modificar qualsevol valor dels atributs de tipus sanció (com per exemple: la regla d'activació). Quan un tipus de sanció no agradi als docents dels centres perquè, per exemple, cap alumne comet aquest tipus de sanció, es farà servir aquest procediment per tal de posar l'atribut ESTAT de la taula tipus de sanció a un estat 'N', no actiu.  
Amb aquest procediment, per tant, es podrà tenir un històric de tots els tipus sancions que han passat pel centre.  
Els paràmetres d'entrada seran tots els atributs de tipus sanció (ldtipussancio, descripció, regla activació, vegades i estat). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
El procediment llançaria un error si l'ldtipussancio posat com a paràmetre d'entrada no existís, perquè no es podrà modificar a un tipus de sanció que no existeix.

### 3.2.2.9 AMONESTACIO

En aquesta taula només es permetrà la inserció, o sigui l'alta de l'amonestació. No es permetrà en cap moment la modificació ni la baixa un cop s'hagi donat d'alta una amonestació. Sabent això només hi haurà un procediment el de l'alta d'amonestacions anomenat SP\_ALTA\_AMONESTACIO. Aquest procediment com s'ha comentat permet donar d'alta les diferents amonestacions que el professorat posarà a l'alumnat. Els seus paràmetres d'entrada són els atributs especificats en la taula amonestació (Idamonestacio, Idalumne, Idprofessor, Idtipusamonestacio, Data hora, comunicat pares). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Per tal de donar d'alta una amonestació el professor té que ser responsable d'algun curs.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos amonestacions no poden tenir dos Idamonestacio iguals ja que cap amonestació pot ser igual a una altra. Aquest camp serà autonumèric.

S'ha de tenir en compte també la seves claus foranes Idalumne, Idprofessor i Idtipusamonestacio en aquest cas si no existeix un alumne amb Alumne.Idalumne = amonestacio.Idalumne o no existeix un professor amb professor.Idprofessor = amonestacio.Idprofessor o tipusamonestacio.Idtipusamonestacio = amonestacio.Idtipusamonestacio també donaria error.

### 3.2.2.10 SANCIO

En aquesta taula només es permetrà la inserció, o sigui l'alta de la sanció. No es permetrà en cap moment la modificació ni la baixa un cop s'hagi donat d'alta una sanció o s'hagi imposat per una acumulació d'amonestacions. Sabent això només hi haurà un procediment el d'alta de sancions anomenat SP\_ALTA\_SANCIO.

Aquest procediment com s'ha comentat permet donar d'alta les diferents sancions que el professorat posarà a l'alumnat. Els seus paràmetres d'entrada són els atributs especificats en la taula sanció (Idsancio, Idalumne, Idtipussancio, resolució i data hora creació). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos sancions no poden tenir dos Idsancio iguals ja que cap sanció pot ser igual a una altra. Aquest camp serà autonumèric.

S'ha de tenir en compte també la seves claus foranes Idalumne i Idtipussancio en aquest cas si no existeix un alumne amb Alumne.Idalumne = sancio.Idalumne o no existeix un tipussancio.Idtipussancio = sancio.Idtipussancio també donaria error.

### 3.2.2.11 TUTOR

En aquesta taula només es permetrà la inserció i la modificació de l'alta d'un tutor de l'alumne. No es permetrà en cap moment la baixa un cop s'hagi donat d'alta el tutor de l'alumne. Sabent això només hi haurà dos procediments el d'alta i modificació de tutors:

- SP\_ALTA\_TUTOR, aquest procediment permet donar d'alta els diferents tutors dels alumnes del centre.  
Els seus paràmetres d'entrada són els atributs especificats en la taula tutor (Idalumne i Idprofessor). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.  
Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Un alumne només podrà tenir un tutor assignat.  
S'ha de tenir en compte també les seves claus foranes Idalumne i Idprofessor, en aquest cas si no existeix un alumne amb alumne.Idalumne = tutor.Idalumne o un professor amb professor.Idprofessor = tutor.idprofessor també donaria error.
- SP\_MOD\_TUTOR, aquest procediment permet modificar el tutor d'un alumne. S'ha considerat que al llarg d'un curs un alumne tindrà un tutor assignat però que aquest tutor pot anar variant durant el curs.

Els paràmetres d'entrada seran tots els atributs del tutor (Idalumne i Idprofessor). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

El procediment llançaria un error si l'Idalumne o l'Idprofessor posats com a paràmetres d'entrada no existissin, perquè no es podrà modificar un tutor si l'alumne el qual volem canviar de tutor no existeix o si el tutor que li volem assignar a l'alumne no existeix.

### 3.2.2.12 DOCÈNCIA

En aquesta taula només es permetrà la inserció, de l'alta d'una docència. No es permetrà en cap moment la baixa, ni la modificació directe d'una docència, però si estarà permès donar de baixa la docència si l'assignatura o el curs associats a ella es donen de baixa. Sabent això només hi haurà un procediment de l'alta de la docència.

Aquest procediment com s'ha comentat permet donar d'alta les diferents docències (anomenades també classes) que els professors impartiran en el centre . Els seus paràmetres d'entrada són els atributs especificats en la taula docència (Idprofessor, Idassignatura i Idcurs). Com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el procediment s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Respecte els possibles errors del procediment es té el no compliment de la restricció de la clau primària. Dos docències no podran fer-se en el mateix curs, donant la mateixa assignatura amb el mateix horari i amb el mateix professor.

S'ha de tenir en compte també la seves claus foranes Idprofessor, Idassignatura i Idcurs en aquest cas si no existeix un professor amb professor.Idprofessor = docencia.Idprofessor o no existeix un assignatura.Idassignatura = docencia.Idassignatura o no existeix curs.Idcurs = docencia.Idcurs també donaria error.

### 3.2.3 Creació dels llistats requerits

A continuació s'exposa els procediments de consulta d'acord amb els requeriments previs per tal de poder donar resposta a les preguntes plantejades.

#### 3.2.3.1 Amonestacions imposades

Aquest procediment mostrarà un llistat de les amonestacions imposades per part de tots els professors.

Aquest procediment s'anomenarà SP\_LLISTAT\_AMONESTACIONS\_IMPOS. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error si encara no hi ha amonestacions imposades.

### **3.2.3.2 Alumnes d'un curs**

Aquest procediment mostrarà un llistat tots els alumnes que hi ha matriculats en un curs en concret.

Aquest procediment s'anomenarà SP\_LLISTAT\_ALUMNES\_CURS. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error si encara no hi ha matriculacions realitzades.

### **3.2.3.3 Amonestacions i sancions disponibles en el centre educatiu**

Aquest procediment mostrarà tots els tipus de sanció i amonestacions que els professors poden imposar als alumnes.

Aquest procediment s'anomenarà SP\_LLISTAT\_TIPUS\_AMONEST\_SAN. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error si no hi ha encara definits ni tipus d'amonestacions ni tipus de sancions.

### **3.2.3.4 Amonestacions per alumne**

Aquest procediment mostrarà el número d'amonestacions que a rebut un alumne en concret, no importarà en el curs en que estigui matriculat.

Aquest procediment s'anomenarà SP\_LLISTAT\_AMON\_ALUMNE. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error si no hi ha alumnes inserits.

### **3.2.3.5 Sancions per alumne i any**

Aquest procediment mostrarà el número de sancions que ha rebut un alumne en un any en concret. En aquest cas com que només es farà per l'any en curs serà en l'any 2010-2011.

Aquest procediment s'anomenarà SP\_LLISTAT\_SANCIONS\_ALUMNE. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error si encara no hi ha alumnes inserits.

### **3.2.3.6 Amonestacions per professor i any**

Aquest procediment mostrarà la mitjana d'amonestacions que tindrà un professor al llarg de tot l'any. En aquest cas en l'any 2010-2011. El procediment s'anomenarà SP\_LLISTAT\_MITJANA\_PROFESSOR. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error si no hi ha encara professor inserits.

### **3.2.3.7 Sancions per a curs i any**

Aquest procediment mostrarà el número de sancions per a un curs en concret i en l'any 2010-2011.

Aquest procediment s'anomenarà SP\_LLISTAT\_SAN\_CURS\_ANY. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error si encara no hi ha cursos inserits.

### **3.2.3.8 Alumne més sancionat en un any**

Aquest procediment mostrarà la llista del nom de l'alumne més sancionat.

Només podrem consultar l'any 2010-2011 que és l'any en curs.

Aquest procediment s'anomenarà SP\_LLISTAT\_ALUMNE\_MES\_SAN. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error quan no hi hagi alumnes inserits.

### **3.2.3.9 Professor més amonestador per curs**

Aquest procediment mostrarà la llista del professor més amonestador per a cada curs.

Aquest procediment s'anomenarà SP\_LLISTAT\_PROFESSOR\_MES\_AMON. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista donarà un error quan no hi hagi professors inserits.

### **3.2.3.10 Mitja de les sancions que tenen els alumnes per curs**

Aquest procediment mostrarà la llista de la mitjana de sancions que rep cada alumne per els diferents cursos.

Aquest procediment s'anomenarà SP\_LLISTAT\_MITJANA\_ALUM\_CURS. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error quan no hi hagi alumnes inserits.

### **3.2.3.11 Alumnes que no tenen cap amonestació**

Aquest procediment mostrarà la dels alumnes que no han rebut encara cap amonestació.

Aquest procediment s'anomenarà SP\_LLISTAT\_ALUM\_CAP\_AMON. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error quan tots els alumnes tinguin almenys una amonestació.

### **3.2.3.12 Llistat de curs i alumne**

Aquest procediment mostrarà la llista dels cursos en els que pertany cada alumne.

Aquest procediment s'anomenarà SP\_LLISTAT\_CURS\_ALUMNE. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error quan no hi hagi alumnes inserits o cursos inserits

### **3.2.3.13 Llistat de professor i assignatura**

Aquest procediment mostrarà la llista de les assignatures dels diferents distribuïdes pel professor responsable (professor que imparteix l'assignatura).

Aquest procediment s'anomenarà SP\_LLISTAT\_DOCENCIA. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista donarà error quan no hi hagi professor inserits o assignatures inserides.

#### **3.2.3.14 Llistat del professor responsable de cada curs**

Aquest procediment mostrarà el professor responsable d'un curs que s'imparteix en el centre.

Aquest procediment s'anomenarà SP\_RESPONABLE\_CURS. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta llista crearà un error quan no hi hagi cursos inserits.

#### **3.2.3.15 Llistat de les hores de consulta als alumnes**

Aquest procediment mostrarà el dia i hora que el professor té consulta per a parlar amb els alumnes.

Aquest procediment s'anomenarà SP\_HORARI\_ATENCIO\_ALUMNES. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta taula crearà un error quan no hi hagi professors inserits.

#### **3.2.3.16 Llistat de les hores de consulta als pares**

Aquest procediment mostrarà el dia i hora que el professor té consulta per a parlar amb els pares.

Aquest procediment s'anomenarà SP\_HORARI\_ATENCIO\_PARES. No es tindran paràmetres d'entrada i com a paràmetre de sortida tindrem la variable RSP de tipus CHAR que donarà el valor 'OK' si el llistat s'ha desenvolupat correctament i donarà 'ERROR: descripció error' quan el procediment hagi causat un error.

Aquesta taula crearà un error quan no hi hagi professors inserits.

#### **3.2.3.17 Llistat de les amonestacions i sancions d'un alumne**

Aquest procediment mostrarà per un alumne en concret les amonestacions i sancions que ha rebut. Aquest té un paràmetre d'entrada per tal de que s'indiqui l'alumne del que volem conèixer aquesta informació.

Aquest procediment s'anomenarà SP\_AMON\_SANCIONS\_ALUMNE (ALUMNE IN CHAR). Com paràmetre d'entrada disposa de lalumne referint-se al DNI de l'alumne del que es vol la informació de les amonestacions i sancions.

Aquesta taula crearà un error quan no hi hagi alumnes inserits.

### 3.3 Creació del joc de proves

Es comença el joc de proves provant els possibles errors que podrien donar els llistats si no hi tinguéssim dades introduïdes.

--LLISTATS SENSE CAP DADA INSERTADA

En cadascuna de les respostes d'aquestes sentències ens surt un avís dient el que falta insertar per donar la llista escollida.

Ara es comença a introduint les dades amb els procediments d'alta i en aquest ordre en concret. Les dades escollides són d'un centre fictici.

- **Tipus d'amonestacions**, en aquest cas tots els tipus d'amonestacions que s'introduiran estaran amb l'atribut ESTAT actiu 'S'. Però es provarà durant el joc de proves les diferents accions que es poden donar quan un tipus d'amonestació passa d'estar en actiu per estar en no actiu. En aquest mateix apartat es provarà també de donar de baixa un tipus d'amonestació amb el procediment de baixa, recordem que aquest només s'utilitza quan s'ha introduït el tipus d'amonestació però s'adonen que mai l'utilitzaran. S'utilitzarà el procediment SP\_MOD\_TIPUS\_AMONESTACIO per a modificar el tipus d'amonestació amb Idamonestació =3.

Com a resultat d'aquestes sentències es tindrà a la taula tipus amonestació un total de 3 tipus diferents d'amonestacions i un, en concret el registre amb Idamonestacio=3 tindrà l'atribut ESTAT='N'.

També es provarà de introduir un tipus d'amonestació ja introduït, amb això s'ha aconseguirà restricció a la clau primària de la taula AMONESTACIO.

- **Tipus sanció**, en aquest cas tots els tipus de sanció que s'introduiran estaran amb l'atribut ESTAT actiu 'S'. Però es provarà durant el joc de proves les diferents accions que es poden donar quan un tipus d'amonestació passa d'estar en actiu per estar en no actiu. En aquest mateix apartat es provarà també de donar de baixa un tipus de sanció amb el procediment de baixa, recordem que aquest només s'utilitza quan s'ha introduït el tipus de sanció però s'adonen que mai l'utilitzaran. S'utilitzarà el procediment SP\_MOD\_TIPUS\_SANCIO per a modificar el tipus de sanció amb Idsancio =3.

Com a resultat d'aquestes sentències es tindrà a la taula tipus amonestació un total de 3 tipus diferents d'amonestacions i un, en concret el registre amb Idamonestacio=3 tindrà l'atribut ESTAT='N'.

També es provarà de introduir un tipus de sanció ja introduïda, amb això s'ha aconseguirà restricció a la clau primària de la taula SANCIO.



- **Professor**, tots els professors introduïts tindran l'atribut ESTAT actiu 'S'. Però es provarà durant el joc de proves les diferents accions que es poden donar quan un professor passa d'estar en actiu per estar en no actiu. En aquest mateix apartat es provarà també de donar de baixa a un professor amb el procediment de baixa, recordem que aquest només s'utilitza quan s'ha introduït el professor però mai comença a donar classe. S'utilitzarà el procediment SP\_MOD\_PROFESSOR per a modificar el professor amb Idprofessor = '11111111W'.

Com a resultat d'aquestes sentències es tindrà a la taula professor un total de 8 professor diferents i un, en concret el registre amb Idprofessor= '11111111W' tindrà l'atribut ESTAT='N'. També es provarà de introduir un professor ja introduït, amb això s'ha aconseguirà restricció a la clau primària de la taula PROFESSOR.

- **Alumne**, tots els alumnes introduïts tindran l'atribut ESTAT actiu 'S'. Però es provarà durant el joc de proves les diferents accions que es poden donar quan un alumne passa d'estar en actiu per estar en no actiu. En aquest mateix apartat es provarà també de donar de baixa a un alumne amb el procediment de baixa, recordem que aquest només s'utilitza quan s'ha introduït l'alumne però mai es matriculat en cap curs. S'utilitzarà el procediment SP\_MOD\_ALUMNE per a modificar l'alumne amb Idalumne = '25656789K'.

Com a resultat d'aquestes sentències es tindrà a la taula alumne un total de 7 alumnes diferents i un, en concret el registre amb Idalumne = '25656789K' tindrà l'atribut ESTAT='N'. També es provarà de introduir un alumne ja introduït, amb això s'ha aconseguirà restricció a la clau primària de la taula ALUMNE. També es provarà d'introduir un alumne amb el mateix DNI que un professor.

- **Assignatures**, en aquest cas les assignatures a insertar tindran un Idassignatura diferent. Les donarem d'alta amb SP\_ALTA\_ASSIGNATURA. Es donarà la assignatura amb Idassignatura=7 de baixa i modificarem la assignatura amb Idassignatura=8.

Com a resultat obtindrem a la taula ASSIGNATURA 7 assignatures diferents. Comprovarem com no podem tenir dos assignatures amb Idassignatura iguals.

- **Calendari**, en aquest cas es donaran d'alta uns tipus de calendari assignats a les assignatures abans inserides. Es tindran calendaris diferents per a les assignatures inserides. Es tractarà de fer un alta d'un calendari en que no existeixi l'assignatura i d'un calendari idèntic a un ja insertat. Amb el procediment de baixa donarem de baixa un calendari inserit prèviament i modificarem un altre.

Com a resultat obtindrem a la taula CALENDARI 6 calendaris diferents. Comprovarem com no podem tenir dos calendaris amb la mateixa clau primària i a més que no podem crear un calendari si no primer no esta inserida l'assignatura a la que es refereix.

- **Curs**, en aquest cas es donarà d'alta uns quants cursos. Farem altes de cursos on el calendari encara no existeix, aquest cas donarà un error advertint-nos d'aquest fet.

També es donarà d'alta un curs on el professor responsable tampoc existeix, aquest cas donarà un error advertint-nos d'aquest fet.

S'intentarà donar d'alta amb un professor responsable que ha sigut donat de baixa (ESTAT='N') i es podrà veure que ens torna a advertir de la situació no deixant en cap cas dels anteriors donar d'alta el curs.

S'intentarà posar en prova que dos cursos tinguin el mateix camp Idcurs (clau primària del curs) i intentar donar d'alta un curs amb un professor inexistent. En tots dos casos la variable de sortida ens donarà l'error.

Donarem de baixa el curs amb Idcurs=2 i modificarem amb un altra professor responsable el curs amb Idcurs=1;

- **Docència**, es donarà d'alta les diferents docències que tindran en el centre. En aquest cas comprovarem que no podem tenir una docència que tingui la mateixa clau primària que una altra, que no podem tenir docència si l'assignatura no esta donada d'alta o si el curs no esta donat d'alta o si el professor que impartirà l'assignatura no esta donat d'alta. També es farà inserir una docència amb un professor que ja no estigui actiu.

S'ha de recordar que un cop s'hagi donat d'alta la docència no és podrà esborrar directament a no ser que s'esborri el curs o l'assignatura del registre.

- **Matriculació**, es donarà d'alta les diferents matriculacions dels diferents alumnes del centre. En aquest cas es comprova que no podem tenir una matriculació que tingui la mateixa clau primària que una altra, fet que porta a la conclusió que cada alumne només podrà matricular-se d'un curs en concret (primer, segon, tercer o quart d'ESO). Es portarà a prova el següent, si hi ha un alumne matriculat en un curs no hi pot matricular-se d'un altra, si no està donat d'alta l'alumne la matricula donarà error i també donarà error si el curs en el que vol matricular-se l'alumne no esta inserit en la taula curs.

S'ha de recordar que un cop s'hagi donat d'alta la matriculació no és podrà esborrar directament a no ser que s'esborri el curs.

- **Amonestació**, només es podrà donar d'alta una amonestació, no es podrà ni esborrar ni modificar. No es podrà crear si l'alumne, el professor o el tipus d'amonestació no existeixen.  
Tampoc es podran donar d'alta si l'alumne o el professor o, fins i tot, el tipus d'amonestació estan en ESTAT='N'.  
Tampoc es podrà donar d'alta si no compleix amb la seva clau primària, dues amonestacions no poden tenir dos Idamonestacio iguals.  
Tampoc es podrà donar d'alta si el tipus amonestació esta en un ESTAT='N'.
- **Sanció**, només es podrà donar d'alta. En aquest lloc de proves només es posarà en proves les sancions per acumulacions d'amonestacions. És veurà que si un alumne arriba tard tres cops, obté tres amonestacions per aquest fet i automàticament obté una sanció. Passa el mateix si un alumne fa cinc vegades soroll a classe, obtindrà cinc amonestacions per aquest fet i automàticament obté una sanció.  
Si volem donar una sanció personalitzada (no per acumulacions) s'ha de tenir en compte que existeixi l'alumne i que a més existeixi el tipus de sanció que li anem a imposar.  
Cap de les sancions imposades podrà tenir un Idsancio igual que una altra.  
Tampoc es podrà donar d'alta si el tipus sanció esta en un ESTAT='N'.

Un cop tenim tot això només quedarà provar d'executar totes les llistes del principi per tal de veure els resultats.

#### 4. Valoració econòmica del projecte

Els recursos necessaris per tal de dur a terme aquest projecte dependran de les condicions que tingui la Generalitat de Catalunya.

Si aquesta disposa d'un hosting propi amb el seu domini propi i el seu parking DNS no serà necessària una inversió addicional en hardware.

Opcionalment, en cas de que el client no disposi d'aquest hosting, seran requerides les inversions en hardware següents:

- Servidor HP ProLiant DL160 G6 E5620 1P, 8 GB-R P410, SAS/SATA, 4 LFF, 500 W, PS (590161-421). Preu 1805 euros.

Amb les següents especificacions:



Especificaciones técnicas	
Procesador	Intel® Xeon® E5620 (12 núcleos, 2,40 GHz, 4 MB L3, 80W)
Número de procesadores	1
Núcleo de procesador disponible	4
Memoria de serie	8 GB
Ranuras de memoria	12 ranuras DIMM
Memoria	PC3-10600R-9
Ranuras de expansión	2
Controlador de red	(1) 2 Puertos 1 GbE NC362i
Tipo de fuente de alimentación	(1) 500 W salida múltiple de alta eficacia
Controlador de almacenamiento	(1) Smart Array P410 SAS/SATA RAID
Software de gestión	N/D
Tipo de unidad óptica	Ningún estándar de suministro
Formato (totalmente configurado)	1U
Software de gestión remota	Lights-Out 100 Advanced
Garantía (Piezas, mano de obra, a domicilio)	3/0/0

- Dos unitats de disc dur HP ENT port doble sense conex. calent (3,5") 600GB 6G SAS 15.000 rpm LFF, (516830-B21). Preu dels dos 540 x 2 = 1080 euros.  
Aquest disc dur és d'alta velocitat per tal de donar un rendiment òptim de la base de dades.



Com que en aquest projecte s'utilitzarà la versió gratuïta d'Oracle (Oracle Express v10.2.00.1) les despeses per software serien de 0 euros.

Honoraris:

L'import total del projecte es desglossa en:

Tasques de disseny

Tasques de programació

Hardware (Opcional)

Les tasques addicionals a realitzar fora del contingut d'aquest projecte es facturaran segons tarifa de 85 euros / hora.

S'estima que les despeses totals del disseny i implementació seran de 9.690 euros.

Concepte	Import
Hardware	Generalitat disposa de Hosting 0 euros
	Necessitat d'un Hosting 2.885 euros
Software	Software lliure 0 euros
Desenvolupament del projecte	9.690 euros

\*Els imports mencionats no inclouen el 18% de l'IVA.

## 5. Conclusions

Com a conclusió es pot dir que els objectius plantejats al llarg del quadrimestre han sigut acabats amb èxit.

Per a fer el projecte, disseny i implementació s'ha pres el model de funcionament d'una de les escoles de la Generalitat. Amb això s'ha aconseguit fer un sistema àgil i amb una bona facilitat de maneig per part de les persones a utilitzar el sistema.

Totes les taules i procediments esmentats són objectes sempre presents en una escola de secundària, tot i això, ha fet falta un estudi profund de la situació a plantejar per tal que compleixi totes les expectatives requerides.

Per últim, el joc de proves ha facilitat la feina de la comprovació de tot el funcionament de la base de dades.

## 6. Glossari

**sistema d'informació** Sistema que recull, emmagatzema i distribueix informació sobre l'estat d'un domini.

**atribut** m Propietat d'una entitat.

**camp** m Representació del valor d'un atribut.

**clau** f Atribut o conjunt d'atributs que permet identificar els objectes (distingir-los els uns dels altres).

**dada** f Nom que rep la informació en el món de les representacions informàtiques.

**entitat** f Conceptualització d'un objecte del món real. El concepte del qual una entitat és instància s'anomena també.

**base de dades (BD)** f Conjunt estructurat de dades que representa entitats i les seves interrelacions. La representació serà única, integrada, malgrat que ha de permetre utilitzacions diverses i simultànies.

**client/servidor (C/S)** m Tecnologia habitual per a distribuir dades. La idea és que dos processos diferents, que es poden executar en un mateix sistema o en sistemes separats, actuen de manera que l'un fa de client o peticionari d'un servei i l'altre fa de servidor. Un procés client pot demanar serveis a diversos servidors. Un servidor pot rebre peticions de molts clients. En general, un procés que fa de client demanant un servei a un altre procés B, pot fer també de servidor d'un servei que li demani un altre procés C.

**clau primària d'una relació** f Clau candidata de la relació que s'ha escollit per a identificar les tuples de la relació.

**clau forana d'una relació** f Subconjunt dels atributs de l'esquema de la relació, CF, tal que existeix una relació S (S no ha de ser necessàriament diferent de R) que té per clau primària CP, i es compleix que, per a tota tupla t de l'extensió de R, els valors per a CF de t són o bé valors nuls, o bé valors que coincideixen amb els valors per a CP d'alguna tupla de S.

**esquema** m Element que agrupa un conjunt de components lògics (taules, vistes, procediments emmagatzemats, etc.).

**PL/PgSQL** m Llenguatge de PostgreSQL equivalent al PSM definit en l'SQL estàndard.  
Nota: La sigla PSM correspon a la denominació anglesa persistent stored module.

**procediment emmagatzemat** m Acció o funció definida per un usuari que proporciona un servei determinat. Un cop creat, es guarda en la base de dades i es tracta com un objecte més d'aquesta.

## 7. **Bibliografia**

- Chuck Murray, *Oracle Database SQL Developer User's Guide* [en línea], [http://download.oracle.com/docs/cd/E12151\\_01/doc.150/e12152.pdf](http://download.oracle.com/docs/cd/E12151_01/doc.150/e12152.pdf).
- Pérez López, César (2008), *Oracle PL/SQL*, Ra M, Librería y Editorial Microinformática.
- Scott Urman, (2001), *Oracle8i: Programación avanzada PL/SQL*, McGraw-Hill / Interamericana de España, S.A.
- Sue Harper, December 2009, *Introducing SQL Developer Data Modeler* [en línea], <http://www.packtpub.com/article/introducing-sql-developer-data-modeler-part1>.
- [http://psoug.org/reference/exception\\_handling.html](http://psoug.org/reference/exception_handling.html).
- Material didàctic de l'assignatura *Bases de dades I*.