

Treball de fi de carrera
Bases de dades relacionals

Sistema de gestió d'amonestacions i sancions en centres educatius

Marta Sánchez José

Enginyeria Tècnica d'Informàtica de Gestió

Consultor: Jordi Ferrer Duran

Data Lliurament: 12/06/2011

ÍNDIX DE CONTINGUTS

ÍNDIX DE CONTINGUTS	2
1. RESUM	5
2. INTRODUCCIÓ	6
2.1. Descripció del TFC	6
2.2. Objectius del TFC	6
2.3. Planificació del projecte	6
2.3.1. Metodologia	6
2.3.2. Distribució de les tasques en les PACs	8
2.3.3. Planificació temporal	9
2.3.4. Anàlisi i control de riscos	9
2.3.5. Desviació sobre la planificació prevista	9
2.4. Recursos necessaris	10
2.5. Productes obtinguts	10
2.6. Breu descripció dels altres capítols de la memòria	10
3. RECOLLIDA I ANÀLISI DE REQUERIMENTS DELS USUARIS	11
3.1. Requisits funcionals	11
3.2. Requisits no funcionals	13
4. DISSENY DE LA BASE DE DADES	14
4.1. Disseny conceptual	14
4.1.1. Diagrama E/R del model conceptual	15
4.1.2. Descripció del model E/R	15
4.1.2.1. Entitats	15
4.1.2.2. Interrelacions	20
4.2. Disseny lògic	21
4.2.1. Model relacional	22
4.2.2. Restriccions d'integritat	25
4.2.3. Regles de negoci	25
4.3. Disseny físic	25
4.3.1. Taules	26
4.3.1.1. Taula CALENDARI	26
4.3.1.2. Taula ASSIGNATURA	26
4.3.1.3. Taula CURS	26
4.3.1.4. Taula ASSIGNATURES_CURS	27
4.3.1.5. Taula CALENDARI_ESCOLAR	27
4.3.1.6. Taula ALUMNE	28
4.3.1.7. Taula MATRICULA	29

4.3.1.8.	Taula PROFESSOR	29
4.3.1.9.	Taula IMPARTIR	30
4.3.1.10.	Taula CALENDARI_TUTORIA	30
4.3.1.11.	Taula REUNIO_ALUMNE	31
4.3.1.12.	Taula REUNIO_PARES	31
4.3.1.13.	Taula TIPUS_AMONESTACIO	31
4.3.1.14.	Taula TIPUS_SANCIO	32
4.3.1.15.	Taula AMONESTACIO	32
4.3.1.16.	Taula SANCIO	34
4.3.1.17.	Taula LOG_TAULA	35
4.3.1.18.	Taula NUMERO_AMONESTACIONS_ALUMNE	35
4.3.1.19.	Taula NUMERO_SANCIONS_ALUMNEANY	35
4.3.1.20.	Taula MITJANA_AMONESTACIONS_PROFESSORANY	35
4.3.1.21.	Taula NUMERO_SANCIONS_CURSANY	35
4.3.1.22.	Taula ALUMNE_MES_SANCIONAT_ANY	36
4.3.1.23.	Taula PROFESSOR_MES_AMONESTADOR_CURS	36
4.3.1.24.	Taula MITJANA_SANCIONS_CURS	36
4.3.2.	Procediments emmagatzemats	36
4.3.2.1.	Alta calendari	36
4.3.2.2.	Baixa calendari	37
4.3.2.3.	Modificació calendari	37
4.3.2.4.	Alta assignatura	37
4.3.2.5.	Baixa assignatura	38
4.3.2.6.	Modificació assignatura	38
4.3.2.7.	Alta curs	39
4.3.2.8.	Baixa curs	39
4.3.2.9.	Modificació curs	39
4.3.2.10.	Alta assignatures curs	40
4.3.2.11.	Alta calendari escolar	40
4.3.2.12.	Baixa calendari escolar	41
4.3.2.13.	Modificació calendari escolar	41
4.3.2.14.	Alta alumne	41
4.3.2.15.	Baixa alumne	42
4.3.2.16.	Modificació alumne	42
4.3.2.17.	Alta matricula	43
4.3.2.18.	Baixa matricula	43
4.3.2.19.	Alta professor	43
4.3.2.20.	Baixa professor	44
4.3.2.21.	Modificació professor	44
4.3.2.22.	Alta impartir	45
4.3.2.23.	Baixa impartir	45
4.3.2.24.	Alta calendari tutoria	45
4.3.2.25.	Baixa calendari tutoria	46
4.3.2.26.	Modificació calendari tutoria	46
4.3.2.27.	Alta reunió alumne	47
4.3.2.28.	Baixa reunió alumne	47
4.3.2.29.	Modificació reunió alumne	47
4.3.2.30.	Alta reunió pares	48
4.3.2.31.	Baixa reunió pares	48
4.3.2.32.	Modificació reunió pares	48
4.3.2.33.	Alta tipus amonestació	49

4.3.2.34.	Baixa tipus amonestació.	49
4.3.2.35.	Modificació tipus amonestació.	50
4.3.2.36.	Alta tipus sanció	50
4.3.2.37.	Baixa tipus sanció.	50
4.3.2.38.	Modificació tipus sanció.	51
4.3.2.39.	Alta amonestació.	51
4.3.2.40.	Baixa amonestació	52
4.3.2.41.	Modificació amonestació	52
4.3.2.42.	Alta sanció	53
4.3.2.43.	Baixa sanció	53
4.3.2.44.	Modificació sanció	54
4.3.2.45.	Consulta curs de l'alumne.	54
4.3.2.46.	Consulta professors de les assignatures	54
4.3.2.47.	Consulta tutor del curs	55
4.3.2.48.	Consulta amonestacions i sancions de l'alumne.	55
4.3.2.49.	Llistat amonestacions imposades.	55
4.3.2.50.	Llistat alumnes del curs	56
4.3.2.51.	Llistat amonestacions i sancions.	56
4.3.2.52.	Llistat amonestacions i sancions de l'alumne.	56
4.3.2.53.	Consulta número d'amonestacions alumne	57
4.3.2.54.	Consulta número de sancions alumne i any	57
4.3.2.55.	Consulta mitjana d'amonestacions professor i any.	57
4.3.2.56.	Consulta número de sancions curs i any.	58
4.3.2.57.	Consulta alumne més sancionat any.	58
4.3.2.58.	Consulta professor més amonestador curs	58
4.3.2.59.	Consulta mitjana de sancions curs	59
5.	CONCLUSIONS	59
6.	GLOSSARI	60
7.	BIBLIOGRAFIA.	60

1. RESUM

Aquest treball consisteix en l'elaboració del disseny i la implementació d'una base de dades relacional per a la gestió d'amonestacions i sancions en centres educatius d'ensenyament secundari de Catalunya. Tot i que el sistema es crea a petició de la Conselleria d'Ensenyament de la Generalitat de Catalunya, aquest serà utilitzat pels instituts.

Amb aquest sistema, els instituts podran gestionar la informació referent als seus alumnes, cursos, professors, amonestacions i sancions. També inclou un mòdul estadístic que permetrà fer consultes relacionades amb les amonestacions i les sancions dels alumnes.

Per a elaborar el treball es passaran per diferents fases:

- 1) Anàlisi de requeriments.
- 2) Elaboració d'un disseny conceptual utilitzant un diagrama d'entitat-relació (E/R).
- 3) Transformació del model conceptual al disseny lògic de la base de dades.
- 4) Implementació de la base de dades utilitzant el sistema gestió de bases de dades Oracle 10g.
- 5) Implementació dels procediments emmagatzemats necessaris per a la manipulació de les dades utilitzant el sistema gestió de bases de dades Oracle 10g.
- 6) Elaboració d'un joc de proves que permeti comprovar el bon funcionament del sistema.

2. INTRODUCCIÓ

2.1. Descripció del TFC

Aquest projecte es desenvolupa degut a la petició realitzada per la Conselleria d'Ensenyament de la Generalitat de Catalunya sol·licitant propostes per el disseny d'una base de dades per emmagatzemar informació sobre les amonestacions i sancions imposades a alumnes dels centres educatius d'ensenyament secundari de Catalunya.

Per a l'elaboració d'aquest projecte, es crearà una base de dades relacional, que consisteix en la definició d'una estructura que permet emmagatzemar tot tipus de dades i manipular-les mitjançant operacions d'actualització i consulta. A més, aquest model permet l'establiment de regles d'integritat que les dades emmagatzemades han de complir.

2.2. Objectius del TFC

Objectius generals:

- Posar en pràctica i ampliar els coneixements adquirits durant els estudis de la carrera, especialment en l'àrea de les bases de dades, mitjançant el desenvolupament d'un projecte informàtic.
- Demostrar la capacitat per realitzar un projecte des del seu inici fins la seva implementació i integració.

Objectius específics:

- Posar en pràctica els coneixements adquirits en les assignatures Bases de dades 1 i Bases de dades 2 aplicant-los a un cas real.
- Conèixer i treballar amb el sistema de gestió de bases de dades Oracle 10g.
- Aprofundir en el coneixement del llenguatge SQL.

2.3. Planificació del projecte

2.3.1. Metodologia

Les etapes en que es descompon el projecte són:

- 1) Elaboració del Pla de treball: Es tracta de descriure el sistema a dissenyar i elaborar la planificació per portar a terme el que es demana.

Tasques:

- Llegir l'enunciat del treball fi de carrera (TFC).
- Redactar els apartats resum, descripció i objectius.
- Descompondre en tasques i subtasques el treball a realitzar per portar a terme el projecte i distribuir aquestes tasques en el temps.
- Redactar l'apartat de planificació del projecte.

2) Recollida i anàlisi dels requeriments dels usuaris: En aquesta etapa es tracta de saber què vol l'usuari que ha encarregat el disseny de la base de dades.

Tasques:

- Buscar informació sobre l'aplicació a fer.
- Definir els requeriments funcionals.
- Definir els requeriments no funcionals.

3) Elaboració del model conceptual: Es tracta d'elaborar un esquema de les dades i de les associacions o interrelacions a partir del coneixement dels requeriments i especificacions d'allò que ha de fer. Per construir el model conceptual s'utilitzarà el model entitat-interrelació (E/R).

Tasques:

- Definir les entitats amb els seus atributs.
- Definir les interrelacions entre entitats.
- Fer el model E/R del model conceptual.

4) Transformació del model conceptual a model lògic: Es tracta de transformar el model conceptual al model relacional obtenint el conjunt de relacions amb els seus atributs, claus primàries i claus foranes. A més es defineixen les restriccions d'integritat.

Tasques:

- Transformar les entitats a relacions.
- Transformar les interrelacions a relacions o incloure els atributs de la clau primària de l'entitat forta com a clau forana de l'entitat dèbil.
- Definir les restriccions d'integritat.

5) Construcció del model físic: Es tracta de transformar el model lògic al model del sistema de gestió de bases de dades (SGBD) amb el qual es desitja implementar la base de dades. En aquest cas, el SGBD serà l'Oracle 10g.

Tasques:

- Implementar les taules, índexs, disparadors, etc., necessaris per a crear la base de dades.
- Implementar els procediments emmagatzemats que permetin l'alta, la baixa i la modificació de les taules creades anteriorment.
- Implementar els procediments emmagatzemats que permetin mostrar els llistats següents:
 - De totes les amonestacions imposades als alumnes.
 - De tots els alumnes d'un curs en concret.
 - De tots els tipus d'amonestacions i sancions disponibles per a aplicar en el centre.
 - De totes les amonestacions i sancions d'un alumne.
- Per a implementar el mòdul estadístic, s'haurà de buscar informació sobre com implementar-lo a partir dels procediments emmagatzemats fets anteriorment, per tal de fer les consultes amb un SELECT basic.
- Implementar els procediments emmagatzemats del mòdul estadístic que han de mostrar les dades següents:
 - El número d'amonestacions per alumne.
 - El número de sancions per alumne i any.
 - La mitjana d'amonestacions per professor i any.
 - El número de sancions per curs i any.
 - El nom de l'alumne més sancionat en un any donat.
 - El nom del professor més amonestador per curs.
 - La mitjana de sancions que tenen els alumnes per curs.
 - El número d'alumnes que no tenen cap amonestació.

- Implementar el procediment emmagatzemat que guardi totes les crides realitzades a procediments emmagatzemats.
 - Redactar la documentació relacionada amb totes les funcionalitats implementades.
- 6) Elaboració del joc de proves: Es tracta d'elaborar un joc de proves que permeti comprovar el bon funcionament del sistema.
- 7) Elaboració de la memòria: Es tracta de finalitzar la redacció de la memòria, una part ja s'haurà fet durant el semestre i s'haurà anat entregant a les pacs.
- Tasques:
- Redactar els apartats que queden: productes obtinguts, de breu descripció dels altres capítols de la memòria, conclusions i línees de futur, glossari, bibliografia i annexos.
 - Revisar la memòria i, si s'escau, modificar els apartats que facin falta.
- 8) Elaboració de la presentació virtual del projecte: Es tracta de fer una presentació utilitzant el programa PowerPoint que mostri una perspectiva general del TFC.
- Tasques:
- Buscar informació sobre com fer documents amb el PowerPoint.
 - Fer el document de presentació del projecte en el programa.
- 9) Lliurament final que inclourà la memòria, la presentació virtual i la base de dades implementada amb l'Oracle 10g.
- 10) Debat virtual: Es tracta de respondre a les preguntes fetes per els membres del Tribunal Virtual sobre el projecte entregat.

2.3.2. Distribució de les tasques en les PACs

Les tasques a realitzar en les etapes descrites en l'apartat anterior, es distribuïran entre els diferents lliuraments de les Pacs:

- PAC 1: Pla de Treball
Es lliuraran la planificació del projecte juntament amb el capítol d'introducció de la memòria, és a dir, de l'etapa 1, elaboració del Pla de treball.
- PAC 2: Disseny de la base de dades
Es lliuraran les tasques pròpies del disseny de la base de dades, és a dir, de les etapes següents:
 - 2) Recollida i anàlisi dels requeriments dels usuaris.
 - 3) Elaboració del model conceptual.
 - 4) Transformació del model conceptual a un model lògic general.
- PAC 3: Implementació de la base de dades
Es lliuraran les tasques pròpies de la implementació de la base de dades, és a dir, de l'etapa 5, disseny físic de la base de dades.
- Lliurament final
Es lliurará tot el treball realitzat al llarg del semestre. A la documentació lliurada en les diferents Pacs s'hi afegirà el resta d'etapes de la metodologia, és a dir, l'elaboració del joc de proves, la presentació virtual i la base de dades en SQL.

2.3.3. Planificació temporal

El quadre següent mostra la planificació temporal del projecte distribuïda en les 15 setmanes de que es disposa per a fer el projecte:

Tasca	Data inici	Data fi	Duració en dies	Duració en hores
Lectura TFC, redacció resum, descripció i objectius	04/03/2011	07/03/2011	4	8
Planificació	08/03/2011	20/03/2011	8	16
Recollida i anàlisi de requeriments	21/03/2011	26/03/2011	6	12
Disseny conceptual	27/03/2011	03/04/2011	8	16
Disseny lògic	04/04/2011	07/04/2011	4	8
Implementació de la creació de taules de la base de dades	08/04/2011	17/04/2011	10	20
Implementació dels procediments emmagatzemats d'alta, baixa i modificació	18/04/2011	23/04/2011	6	12
Implementació dels procediments emmagatzemats dels llistats	24/04/2011	30/04/2011	7	14
Implementació dels procediments emmagatzemats del mòdul estadístic	01/05/2011	15/05/2011	15	30
Joc de proves	16/05/2011	23/05/2011	8	16
Redacció i revisió de la memòria	24/05/2011	28/05/2011	5	10
Presentació virtual	29/05/2011	12/06/2011	15	30
Debat Tribunal Virtual	20/06/2011	26/06/2011	7	14
TOTAL:			103	206

Es dedicaran aproximadament 2 hores diàries al projecte.

2.3.4. Anàlisi i control de riscos

En el temps que duri el projecte poden produir alguns riscos generals que poden fer variar la planificació inicial. Per això s'ha de controlar i intentar que el seu impacte sigui mínim.

Els riscos que poden sorgir amb les seves solucions són:

- Desconeixement dels programes Oracle 10g i el PowerPoint: Abans de realitzar les tasques que necessiten aquests programes, es dedicarà un temps a l'estudi d'aquests programes
- Problemes de salut o laborals que impedeixin no poder dedicar temps al projecte durant alguns dies: S'incrementarien les hores a dedicar al projecte, sobretot caps de setmana i festius, per tal de recuperar els dies perduts. Si coincidís amb l'entrega d'una pac i no es pogués realitzar tota la feina proposada en la pac, s'entregaria la part feta i es desplaçaria la part no realitzada per a entregar a la següent pac.

2.3.5. Desviació sobre la planificació prevista

Durant el transcurs del projecte, s'han produït algunes modificacions sobre la planificació prevista al principi del curs.

Aquestes modificacions s'han degut principalment a la falta de previsió sobre les correccions de les pacs entregades anteriorment, provocant que tant la Pac 2 com la Pac 3 s'entreguessin amb uns dies de retard. Per a realitzar el lliurament final s'han ampliat les hores previstes a la planificació en 4 hores setmanals, és a dir, de 14 hores setmanals previstes al principi es van passar a dedicar 18 hores setmanals en les últimes 4 setmanes. De manera, que en total s'han dedicat 222 hores a realitzar el projecte.

2.4. Recursos necessaris

Per al desenvolupament d'aquest projecte s'han utilitzat una sèrie de recursos, tant de maquinari com de programari.

El recurs de maquinari que s'ha utilitzat és un ordinador amb Intel Core 2 DUO CPU T5450 de 1.66 GHz amb una capacitat de disc dur de 250 GB i memòria RAM de 2 GB.

Els recursos de programari utilitzats són:

- Windows Vista Home Premium Service Pack 2.
- Oracle 10g Express Edition, versió 10.2.0.1 (subministrada per la UOC).
- SQL Developer, versió 1.0.0.15 (subministrada per la UOC).
- UML Pad, versió 3.1.1, per a l'elaboració del diagrama E/R del model conceptual.
- Microsoft Office Word 2003, per a l'elaboració de tota la documentació escrita (pacs, memòria , etc).
- Microsoft Office Powerpoint 2003, per a la presentació de la memòria.
- Adobe Reader, versió 8.2.1, per a la lectura de la documentació necessària, en format “.pdf”.
- Internet Explorer 8, navegador per a les consultes realitzades a través d'Internet.

2.5. Productes obtinguts

Durant la elaboració d'aquest projecte de final de carrera s'han generat els següents documents i arxius:

- Els documents que formen les pacs entregades al llarg del semestre: Pac 1, pac 2 i pac 3.
- Scripts de creació de taules, seqüències i disparadors.
- Scripts de creació de procediments.
- Scripts d'inserció de dades.
- Script del joc de proves.
- Memòria final del projecte.
- Presentació virtual del projecte.

2.6. Breu descripció dels altres capítols de la memòria

Els altres capítols de la memòria descriuen el procés realitzat per a arribar a la implementació del producte final.

Per a realitzar-lo s'ha utilitzat el cicle clàssic de desenvolupament de programari que es detallen en els capítols següents: anàlisi, disseny i implementació del producte final.

3. RECOLLIDA I ANÀLISI DE REQUERIMENTS DELS USUARIS

La recollida de requeriments tracta d'obtenir informació sobre els aspectes essencials de la base de dades a desenvolupar: les dades, les funcionalitats i la manera en com interactuaran aquests amb els usuaris. Per a recollir aquesta informació, els dissenyadors de la base de dades s'entrevisten amb els futurs usuaris de la base de dades per a recollir i documentar les seves necessitats d'informació.

Un cop es tenen els requeriments, s'analitzen per a definir un conjunt d'especificacions de requisits; a partir del qual, es realitzaran el dissenys conceptual, lògic i físic que portaran a la definició de les taules que suportaran les dades.

En aquest cas, no es poden entrevistar als usuaris, per tant, la informació que necessitem s'obtindrà de diferents formes:

- De l'enunciat del treball de fi de carrera (TFC).
- Buscant informació relacionada a amonestacions i sancions a Internet.

3.1. Requisits funcionals

Gestió d'alumnes:

De cada alumne guardarem el seu número d'expedient, les seves dades personals (DNI, nom, cognom1, cognom2, data de naixement) i les seves dades de contacte (adreça, ciutat, codi postal, província, telèfon, mòbil, e-mail).

Els alumnes podran ser donats d'alta i de baixa. També es podran modificar totes les seves dades, excepte el seu document nacional d'identitat (DNI).

Gestió de professors:

De cada professor guardarem les seves dades personals (DNI, nom, cognom1, cognom2, data de naixement) i les seves dades de contacte (adreça, ciutat, codi postal, província, telèfon, mòbil, e-mail).

Els professors podran ser donats d'alta i de baixa. També es podran modificar totes les seves dades, excepte el seu document nacional d'identitat (DNI).

Els professors que siguin tutors tindran un horari assignat per atendre als pares i als alumnes. De cada reunió, s'haurà de guardar l'alumne, la data, l'hora i el tema tractat.

Gestió de cursos:

De cada curs guardarem un identificador, el seu nom, l'any que s'imparteix i el tutor. En cada curs s'impartiran varies assignatures.

Els cursos podran ser donats d'alta i de baixa. També es podran modificar totes les seves dades, excepte el seu identificador.

Gestió d'assignatures:

De cada assignatura guardarem un identificador, el seu nom, el curs o cursos on pertanyen i els professors que l'imparteixen.

Les assignatures podran ser donades d'alta i de baixa. També es podran modificar totes les seves dades, excepte el seu identificador.

Gestió del calendari escolar:

De cada calendari escolar guardarem el curs, l'assignatura i l'horari de l'assignatura. Dels calendaris escolars es podran fer altes, baixes i modificacions.

Gestió d'amonestacions:

La gestió d'amonestacions es divideix en dues parts:

- Gestió dels tipus d'amonestacions: Per a cada tipus d'amonestació guardarem un identificador, el tipus d'amonestació i la descripció.
- Gestió d'amonestacions als alumnes: Per a cada amonestació guardarem l'alumne amonestat, la data, l'hora, el curs, l'assignatura, el professor responsable, el tipus d'amonestació, la gravetat i si s'ha comunicat als pares.

D'aquestes dues gestions d'amonestacions es podran fer altes, baixes i modificacions.

Gestió de sancions:

La gestió de sancions es divideix en dues parts:

- Gestió dels tipus de sancions: Per a cada tipus de sanció guardarem un identificador, el tipus de sanció i el moment en que s'activa. Cada sanció anirà lligada a una amonestació, per exemple: la sanció "prova d'avaluació extra" anirà lligada a l'amonestació "alumne soroll".
- Gestió de sancions als alumnes: Per a cada sanció es guardarem l'alumne sancionat, el curs, la data, el tipus de sanció, el motiu i la resolució de la sanció.

D'aquestes dues gestions d'amonestacions es podran fer altes, baixes i modificacions.

Taula log:

Cada vegada que s'executi un procediment emmagatzemat, s'emmagatzemarà en una taula log: el nom del procediment executat, la data i l'hora d'execució i els paràmetres d'entrada i sortida.

Consultes:

- Consulta per saber a quin curs pertany un alumne mostrant l'identificador i el nom del curs.
- Consulta per saber quins professors donen una assignatura mostrant el número de DNI, nom, cognom1 i cognom2.
- Consulta per saber quin professor és el tutor d'un curs mostrant el número de DNI, nom, cognom1 i cognom2 del professor.
- Consulta de totes les amonestacions i sancions d'un alumne mostrant els identificadors, els tipus i les dates de les amonestacions i les sancions.

Llistats:

- Llistat de totes les amonestacions imposades als alumnes mostrant l'identificador, el tipus i la data de les amonestacions, el nom del curs, l'any acadèmic, el DNI, el nom, cognom1 i cognom2 de l'alumne.
- Llistat de tots els alumnes d'un curs en un any acadèmic determinat mostrant el número de DNI, nom, cognom1 i cognom2 dels alumnes.
- Llistat de tots els tipus d'amonestacions i sancions disponibles per a aplicar en el centre mostrant l'identificador, el tipus i la descripció de les amonestacions i l'identificador, el tipus i el moment d'activació de les sancions.
- Llistat de totes les amonestacions i sancions d'un alumne mostrant l'identificador, el tipus, i la data de les amonestacions i de les sancions.

Consultes del mòdul estadístic:

- Consulta del número d'amonestacions per alumne sense tenir en compte el curs de que estigui matriculat.
- Consulta del número de sancions per alumne i any.
- Consulta de la mitjana d'amonestacions per professor i any.
- Consulta del número de sancions per curs i any.
- Donat un any, mostrar l'alumne més sancionat en aquell any, mostrant el número de DNI, nom, cognom1 i cognom2 de l'alumne.
- Consulta del professor que posa més amonestacions per curs, mostrant el número de DNI, nom, cognom1 i cognom2 del professor.
- Consulta de la mitjana de sancions que tenen els alumnes per curs.
- Consulta del número d'alumnes que no tenen cap amonestació.

3.2. Requisits no funcionals

Interfície d'usuari

El disseny d'aquest sistema no contempla el disseny d'una interfície gràfica per a l'usuari final del producte.

Per accedir a la base de dades s'utilitzarà SQL Developer cridant els procediments emmagatzemats creats o utilitzant la clàusula *SELECT* per visualitzar continguts.

Comportament

El sistema haurà de tenir un comportament estable, informant a l'usuari si l'execució ha finalitzat amb èxit o si ha fracassat mostrant el tipus d'error produït.

Cada vegada que s'executi un procediment emmagatzemat, s'emmagatzemarà en una taula log: el nom del procediment executat, la data i l'hora, els paràmetres d'entrada i els de sortida.

Usuaris

Tindrem quatre perfils d'usuari:

- Administrador de la base de dades: Tindrà permisos il·limitats sobre la base de dades per poder fer el manteniment de la base de dades: creació de taules, modificació de taules, inserció de dades, modificació de dades, etc.
- Personal d'administració: S'ocuparà de la gestió dels alumnes, professors, cursos, assignatures i calendari escolar podent fer les funcionalitats d'alta, baixa i modificació d'aquestes.
- Professors: S'ocuparan de la gestió de les amonestacions i de les sancions podent fer les funcionalitats d'alta, baixa i modificació d'aquestes. També podran amonestar i sancionar als alumnes.

Programari

El sistema gestor de bases de dades serà l'Oracle i l'accés es farà utilitzant SQL Developer.

4. DISSENY DE LA BASE DE DADES

4.1. Disseny conceptual

El disseny conceptual conté una descripció detallada dels requeriments d'informació dels usuaris, i conté les descripcions dels tipus de dades, les relacions entre ells, els atributs i les restriccions. Per fer aquest disseny no és té en compte la tecnologia que s'utilitzarà per a fer la implementació.

Per realitzar el disseny conceptual s'ha escollit el model entitat/relació (E/R) que descriu les dades com a entitats, interrelacions i atributs.

Una entitat és qualsevol objecte del món real amb existència pròpia, sobre el qual volem tenir informació en una base de dades. Per exemple: cotxes, cases, treballadors, clients, empreses, etc. Cada entitat té propietats específiques, anomenades atributs que la descriuen. Per exemple, l'entitat PERSONA es pot descriure per dni, nom, cognom1, cognom2, etc. D'aquests atributs n'hi ha com a mínim un que permet diferenciar unes entitats d'altres, no prenen mai el mateix valor per dues entitats; aquest atribut s'anomenen les claus de l'entitat.

En el diagrama E/R, les entitats es representen mitjançant un rectangle i dins es posa el seu nom. Els atributs es representen per el·lipses connectades a l'entitat mitjançant una línia recta i dins es posa el seu nom. Per a diferenciar els atributs que formen la clau dels altres, aquests es posaran en majúscules o es subratllaran.

Una interrelació és una correspondència, associació o connexió entre dues o més entitats. Cada relació pot tenir atributs descriptius.

En les interrelacions hi poden participar dues o més entitats, per exemple les relacions poden ser binàries, ternàries, etc. El nombre d'entitats que hi participen s'anomena grau d'una relació.

La correspondència de cardinalitat expressa el nombre màxim d'entitats que estan relacionades amb una única entitat de l'altre conjunt d'entitats que intervé a la relació. La cardinalitat de les relacions pot ser:

- 1:1 Una a una: La cardinalitat màxima en ambdues direccions és 1.
- 1:N Una a moltes. La cardinalitat màxima en una direcció és 1 i a l'altra moltes.
- N:M Moltes a moltes. La cardinalitat màxima en ambdues direccions és moltes.

En el diagrama E/R, les interrelacions es representen mitjançant un rombe i els seus noms descriuen la seva funció.

En el disseny, també es farà ús de la generalització/especialització, que permet reflectir el fet que hi ha una entitat general, que s'anomena entitat superclasse, que es pot especialitzar en entitats subclasse:

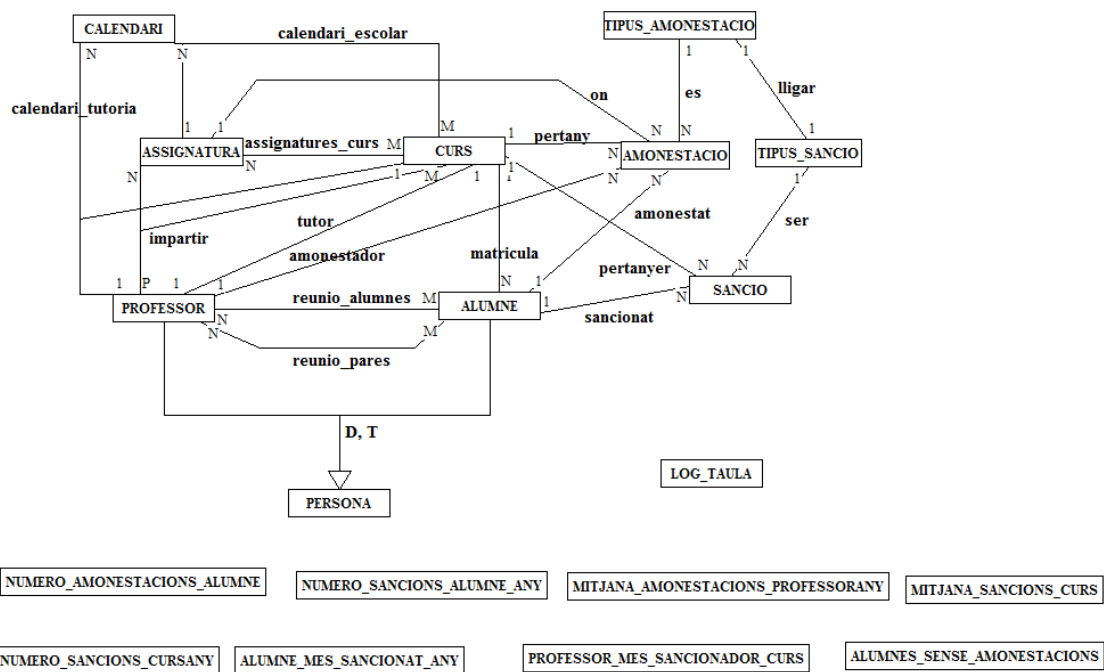
- Entitat superclasse: Permet modelitzar les característiques comunes de l'entitat vista a un nivell genèric.
- Entitats subclasse: Permeten modelitzar les característiques pròpies de les seves especialitzacions. Tota ocurrència d'una entitat subclasse ha de ser també una ocurrència de la seva entitat superclasse.

La generalització/especialització pot ser:

- Disjunta: Una mateixa ocurrència no pot aparèixer a dues entitats subclasse diferents.
- Encavalcada: Una mateixa ocurrència pot aparèixer a dues entitats subclasse diferents.
- Total: Tota ocurrència de l'entitat superclasse ha de pertànyer a alguna de les entitats subclasse.
- Parcial: Tota ocurrència de l'entitat superclasse no cal que pertanyi a una entitat subclasse.

En el diagrama E/R, la generalització/especialització es representa amb fletxes que surten de les entitats subclasse i que es dirigeixen a l'entitat superclasse. Per representar el tipus de generalització/especialització s'escriu a un costat de les fletxes les lletres: "D" si és disjunta, "S" si és encavalcada, "T" si és total i "P" si és parcial.

4.1.1. Diagrama E/R del model conceptual



4.1.2. Descripció del model E/R

Per fer el diagrama del model E/R s'ha utilitzat l'eina UML Pad. Per aquest motiu, les interrelacions no estan senyalades amb rombes, com demana el model E/R, només estan senyalades amb fletxes.

Per facilitar la llegibilitat del model E/R s'ha optat per no incloure els atributs de les entitats i les interrelacions, els quals es mostraran a continuació.

4.1.2.1. Entitats

CALENDARI:

Guarda els dies que formen el calendari del curs, especificant per a cada data el dia de la setmana en que cau i si és festiu.

Els atributs de l'entitat CALENDARI són:

- **Data:** Data del dia, és a dir, dia, mes i any. Camp amb format data. No pot ser nul.
- **DiaSetmana:** Dia de la setmana. Camp alfanumèric de longitud 8. No pot ser nul. Només pot prendre els valors “dilluns”, “dimarts”, “dimecres”, “dijous” i “divendres”.
- **Festiu:** Indica si el dia indicat per l'atribut data és festiu o no. Camp alfanumèric de longitud 1. No pot ser nul. Només pot prendre els valors “s” (si) i “n” (no).

La clau primària de l'entitat està formada per l'atribut *Data*..

ASSIGNATURA:

Guarda les assignatures que s'imparteixen en el centre i formen part d'algun curs.

Els atributs de l'entitat ASSIGNATURA són:

- **IdAssignatura:** Identificador de l'assignatura. Camp numèric de longitud 5. No pot ser nul.
- **Nom:** Nom del curs. Camp alfanumèric de longitud 20. No pot ser nul.

La clau primària de l'entitat està formada per l'atribut *IdAssignatura*.

CURS:

Guarda els cursos que s'imparteixen en el centre.

Els atributs de l'entitat CURS són:

- **IdCurs:** Identificador del curs. Camp numèric de longitud 5. No pot ser nul.
- **AnyAcademic:** Any en que s'inicia el curs, per exemple si es tracta del curs escolar que va de setembre del 2010 al juny del 2011, *AnyAcademic* tindrà el valor 2010. Camp numèric de longitud 4. No pot ser nul.
- **Nom:** Nom del curs. Camp alfanumèric de longitud 20. No pot ser nul.

La clau primària de l'entitat està formada per els atributs *IdCurs* i *AnyAcademic*.

PERSONA:

Com que les dades personals i de contacte són les mateixes tant per a alumnes com per a professors, s'ha optat per fer una superclasse amb els atributs comuns de les dues entitats.

Els atributs de l'entitat PERSONA són:

- **DNI:** DNI de la persona. Camp alfanumèric de longitud 10. No pot ser nul.
- **Nom:** Nom de la persona. Camp alfanumèric de longitud 30. No pot ser nul.
- **Cognom1:** Primer cognom de la persona. Camp alfanumèric de longitud 40. No pot ser nul.
- **Cognom2:** Segon cognom de la persona. Camp alfanumèric de longitud 40. No pot ser nul.
- **DataNaixement:** Data de naixement de la persona. Camp amb format data. No pot ser nul·la.
- **Adreça:** Adreça on està domiciliada la persona. Camp alfanumèric de longitud 50. No pot ser nul·la.
- **Ciutat:** Ciutat on està domiciliada la persona. Camp alfanumèric de longitud 30. No pot ser nul·la.
- **CodiPostal:** Codi postal de la ciutat on està domiciliada la persona. Camp numèric de longitud 5. No pot ser nul.
- **Província:** Província de la ciutat on està domiciliada la persona. Camp alfanumèric de longitud 30. No pot ser nul·la.
- **Telefon:** Número de telèfon de la persona. Camp numèric de longitud 9. Pot ser nul.
- **Mobil:** Número de telèfon mòbil de la persona. Camp numèric de longitud 9. Pot ser nul.
- **Email:** Adreça de correu electrònic de la persona. Camp alfanumèric de longitud 20. Pot ser nul.

La clau primària de l'entitat està formada per l'atribut *DNI*.

ALUMNE:

És una subclasse de PERSONA i tindrà tots els atributs de la classe PERSONA més atributs propis.

Els atributs de l'entitat ALUMNE (atributs no inclosos en l'entitat PERSONA) són:

- **Expedient:** Número d'expedient de l'alumne. Camp alfanumèric de longitud 8. No pot ser nul.

La clau primària de l'entitat està formada per l'atribut *DNI* que hereta de l'entitat superclasse PERSONA. L'atribut *expedient* és clau alternativa de l'entitat.

PROFESSOR:

És una subclasse de PERSONA i tindrà tots els atributs de la classe PERSONA més atributs propis.

Tots els atributs de l'entitat PROFESSOR estan inclosos l'entitat PERSONA).

La clau primària de l'entitat està formada per l'atribut *DNI* que hereta de l'entitat superclasse PERSONA.

TIPUS_AMONESTACIO:

Guarda els diferents tipus d'amonestacions.

Els atributs de l'entitat TIPUS_AMONESTACIO són:

- **IdTipus:** Identificador del tipus d'amonestació. Camp numèric de longitud 5. No pot ser nul.
- **Tipus:** Tipus d'amonestació. Camp alfanumèric de longitud 30. No pot ser nul.
- **Descripció:** Descripció de l'amonestació. Camp alfanumèric de longitud 50. No pot ser nul.

La clau primària de l'entitat està formada per l'atribut *IdTipus*.

L'atribut *tipus* és clau alternativa de l'entitat.

TIPUS_SANCIO:

Guarda els diferents tipus de sancions.

Els atributs de l'entitat TIPUS_SANCIO són:

- **IdTipus:** Identificador del tipus de sanció. Camp numèric de longitud 5. No pot ser nul.
- **Tipus:** Tipus de sanció. Camp alfanumèric de longitud 60. No pot ser nul.
- **Activació:** Moment en que s'activa la sanció. Camp alfanumèric de longitud 60. No pot ser nul.
- **Valor:** Valor numèric definit per a poder llançar automàticament una sanció. Camp numèric de longitud 2. No pot ser nul.

La clau primària de l'entitat està formada per l'atribut *IdTipus*.

L'atribut *tipus* és clau alternativa de l'entitat.

AMONESTACIO:

Guarda les amonestacions imposades als alumnes.

Els atributs de l'entitat AMONESTACIO són:

- **IdAmonestacio:** Identificador de l'amonestació. Camp numèric de longitud 5. No pot ser nul.
- **Data:** Data en que es posa l'amonestació. Camp amb format data. No pot ser nul.
- **Hora:** Hora en que es posa l'amonestació. Camp amb format hora. No pot ser nul.
- **ComunicacioPares:** Indica si els pares han estat informats de l'amonestació rebuda pel fill. Camp alfanumèric de longitud 1. No pot ser nul. Només pot prendre els valors "s" (si) i "n" (no).
- **Gravetat:** Gravetat de l'amonestació. Camp alfanumèric de longitud 10. No pot ser nul.

La clau primària de l'entitat està formada per l'atribut *IdAmonestacio*.

SANCIO:

Guarda les sancions imposades als alumnes.

Els atributs de l'entitat SANCIO són:

- **IdSancio:** Identificador de la sanció. Camp numèric de longitud 5. No pot ser nul.
- **Data:** Data que es posa la sanció. Camp amb format data. No pot ser nul.
- **Motiu:** Motiu de la sanció. Camp alfanumèric de longitud 50. No pot ser nul.
- **Resolució:** Resolució de la sanció. Camp alfanumèric de longitud 50. No pot ser nul.

La clau primària de l'entitat està formada per l'atribut *IdSancio*.

LOG_TAULA:

Guarda les dades relacionades amb els accessos a la base de dades.

Els atributs de l'entitat LOG_TAULA són:

- **Procediment:** Nom del procediment executat. Camp alfanumèric de longitud 30. No pot ser nul.
- **Data:** Data de l'execució del procediment. Camp amb format data. No pot ser nul.
- **Hora:** Hora de l'execució del procediment. Camp amb format hora. No pot ser nul.
- **Entrada:** Paràmetres d'entrada al procediment. Camp alfanumèric de longitud 512. No pot ser nul.
- **Sortida:** Paràmetres de sortida del procediment. Camp alfanumèric de longitud 512. No pot ser nul.

La clau primària de l'entitat està formada per els atributs *Procediment*, *Data* i *Hora*.

NUMERO_AMONESTACIONS_ALUMNE:

Aquesta taula forma part del mòdul estadístic i guarda les dades dels alumnes amb el número total d'amonestacions que li han sigut imposades, independentment del curs.

Els atributs de l'entitat NUMERO_AMONESTACIONS_ALUMNE són:

- **DNI:** DNI de l'alumne. Camp alfanumèric de longitud 10. No pot ser nul.
- **NumAmonestacions:** Número d'amonestacions de l'alumne. Camp numèric de longitud 3. No pot ser nul.

NUMERO_SANCIONS_ALUMNEANY:

Aquesta taula forma part del mòdul estadístic i guarda les dades dels alumnes amb el número de sancions que li han sigut imposades cada any acadèmic.

Els atributs de l'entitat NUMERO_SANCIONS_ALUMNEANY són:

- **DNI:** DNI de l'alumne. Camp alfanumèric de longitud 10. No pot ser nul.
- **AnyAcademic:** Any en que s'inicia el curs. Camp numèric de longitud 4. No pot ser nul.
- **NumSancions:** Número de sancions de l'alumne. Camp numèric de longitud 3. No pot ser nul.

MITJANA_AMONESTACIONS_PROFESSORANY:

Aquesta taula forma part del mòdul estadístic i guarda les dades dels professors amb la mitjana d'amonestacions que ha imposat cada any acadèmic.

Els atributs de l'entitat MITJANA_AMONESTACIONS_PROFESSORANY són:

- **DNI:** DNI del professor. Camp alfanumèric de longitud 10. No pot ser nul.
- **AnyAcademic:** Any en que s'inicia el curs. Camp numèric de longitud 4. No pot ser nul.
- **MitjanaAmonestacions:** Mitjana d'amonestacions del professor. Camp numèric de longitud 3. No pot ser nul.
- **NumAmonestacions:** Número d'amonestacions del professor. Camp numèric de longitud 3. No pot ser nul.

NUMERO_SANCIONS_CURSANY:

Aquesta taula forma part del mòdul estadístic i guarda les dades dels cursos amb el número de sancions que li han sigut imposades cada any acadèmic.

Els atributs de l'entitat NUMERO_SANCIONS_CURSANY són:

- **Curs:** Identificador del curs. Camp numèric de longitud 5. No pot ser nul.
- **AnyAcademic:** Any en que s'inicia el curs. Camp numèric de longitud 4. No pot ser nul.
- **NumSancions:** Número de sancions del curs. Camp numèric de longitud 3. No pot ser nul.

ALUMNE_MES_SANCIONAT_ANY:

Aquesta taula forma part del mòdul estadístic i guarda les dades de l'alumne més sancionat per any acadèmic.

Els atributs de l'entitat ALUMNE_MES_SANCIONAT_ANY són:

- **DNI:** DNI de l'alumne. Camp alfanumèric de longitud 10. No pot ser nul.
- **AnyAcademic:** Any en que s'inicia el curs. Camp numèric de longitud 4. No pot ser nul.
- **NumSancions:** Número de sancions de l'any. Camp numèric de longitud 3. No pot ser nul.

PROFESSOR_MES_AMONESTADOR_CURS:

Aquesta taula forma part del mòdul estadístic i guarda les dades del professor més amonestador per curs.

Els atributs de l'entitat PROFESSOR_MES_AMONESTADOR_CURS són:

- **DNI:** DNI del professor. Camp alfanumèric de longitud 10. No pot ser nul.
- **Curs:** Identificador del curs. Camp numèric de longitud 5. No pot ser nul.
- **AnyAcademic:** Any en que s'inicia el curs. Camp numèric de longitud 4. No pot ser nul.
- **NumAmonestacions:** Número d'amonestacions per curs. Camp numèric de longitud 3. No pot ser nul.

MITJANA_SANCIONS_CURS:

Aquesta taula forma part del mòdul estadístic i guarda les dades dels cursos amb la mitjana de sancions d'alumnes.

Els atributs de l'entitat MITJANA_SANCIONS_CURS són:

- **Curs:** Identificador del curs. Camp numèric de longitud 5. No pot ser nul.
- **AnyAcademic:** Any en que s'inicia el curs. Camp numèric de longitud 4. No pot ser nul.
- **MitjanaSancions:** Mitjana de sancions del professor. Camp numèric de longitud 3. No pot ser nul.
- **NumSancions:** Número de sancions del professor. Camp numèric de longitud 3. No pot ser nul.

ALUMNES_SENSE_AMONESTACIONS:

Aquesta taula forma part del mòdul estadístic i guarda les dades dels alumnes que no tenen cap amonestació.

Els atributs de l'entitat ALUMNES_SENSE_AMONESTACIONS són:

- **DNI:** DNI de l'alumne. Camp alfanumèric de longitud 10. No pot ser nul.

4.1.2.2. Interrelacions

CALENDARI_ESCOLAR:

Interrelació que representa el calendari escolar. Està formada per una interrelació ternària entre les entitats CALENDARI, ASSIGNATURA i CURS.

Els atributs d'aquesta interrelació són:

- **HoraInici:** Hora inici que forma part de l'horari. Camp alfanumèric de longitud 5. No pot ser nul.
- **HoraFi:** Hora final que forma part de l'horari. Camp alfanumèric de longitud 5. No pot ser nul.

CALENDARI_TUTORIA:

Interrelació que indica quin és l'horari dels tutors per a atendre tant les consultes dels alumnes com les dels pares dels alumnes. Està formada per una interrelació ternària entre les entitats CALENDARI, PROFESSOR i CURS.

Els atributs d'aquesta interrelació són:

- **HoraInici:** Hora inici que forma part de l'horari. Camp alfanumèric de longitud 5. No pot ser nul.
- **HoraFi:** Hora final que forma part de l'horari. Camp alfanumèric de longitud 5. No pot ser nul.
- **HorariAlumnes:** Indica si l'horari de consulta és per atendre als alumnes o als pares. Camp alfanumèric de longitud 1. No pot ser nul. Només pot prendre els valors "a" (alumnes) i "p" (pares).

IMPARTIR:

Interrelació que indica les assignatures que imparteix un professor. Està formada per una interrelació ternària entre les entitats ASSIGNATURA, PROFESSOR i CURS.

TUTOR:

Interrelació que indica de quin curs és tutor el professor.

AMONESTADOR:

Interrelació que indica el professor que ha posat cada amonestació.

REUNIO_ALUMNE:

Interrelació que indica les reunions de consulta realitzades entre professor i alumne.

Els atributs d'aquesta interrelació són:

- **Data:** Data de la reunió. Camp amb format data. No pot ser nul.
- **Hora:** Hora de la reunió. Camp amb format hora. No pot ser nul.
- **Tema:** Tema tractat en la reunió. Camp alfanumèric de longitud 50. Pot ser nul.

REUNIO_PARES:

Interrelació que indica les reunions realitzades entre un professor i els pares d'un alumne.

Els atributs d'aquesta interrelació són:

- **Data:** Data de la reunió. Camp amb format data. No pot ser nul.
- **Hora:** Hora de la reunió. Camp amb format hora. No pot ser nul.
- **Tema:** Tema tractat en la reunió. Camp alfanumèric de longitud 50. Pot ser nul.

ASSIGNATURES_CURS:

Interrelació que indica les assignatures que te cada curs.

MATRICULA:

Interrelació que indica a quin curs està matriculat cada alumne.

AMONESTAT:

Interrelació que indica quin alumne ha estat amonestat.

PERTANY:

Interrelació que indica de quin curs és l'alumne amonestat.

ON:

Interrelació que indica quina assignatura que estava fent l'alumne quan ha estat amonestat.

ES:

Interrelació que indica de quin tipus és l'amonestació.

SANCIONAT:

Interrelació que indica quin alumne ha estat sancionat.

PERTANY:

Interrelació que indica de quin curs és l'alumne sancionat.

SER:

Interrelació que indica de quin tipus és la sanció.

LLIGAR:

Interrelació que indica el lligam entre les sancions i les amonestacions, per exemple l'amonestació "alumne_arriba_tard" està lligada a la sanció "quedar-se una hora extra d'estudi durant una setmana".

Els atributs d'aquesta interrelació són:

- **Valor:** Data de la reunió. Camp numèric de longitud 2. No pot ser nul.

Generalització/especialització:

La generalització/especialització és:

- Disjunta: Una persona no pot ser alumne i professor alhora.
- Total: Tota persona serà o bé alumne o bé professor.

4.2. Disseny lògic

En aquesta fase partim del resultat anterior, del disseny conceptual.

El disseny lògic converteix el disseny conceptual en un disseny relacional que s'ajusti al model del sistema gestor de bases dades on s'implementarà la base de dades.

Les entitats del model E/R originaran relacions i les interrelacions poden donar lloc a claus foranes d'alguna relació ja obtinguda o poden donar lloc a una nova relació.

Per la transformació de les interrelacions binàries es té en compte la seva connectivitat:

- Connectivitat 1:1: Cal afegir a qualsevol de les dues relacions una clau forana que faci referència l'altra relació.

- Connectivitat 1:N: Cal afegir a la relació corresponent a l'entitat del costat N una clau forana que faci referència a l'altra relació.
- Connectivitat N:M: Es crea una relació on la seva clau primària estarà formada pels atributs de la clau primària de les dues entitats interrelacionades. Els atributs de la interrelació seran atributs de la nova relació.

Les interrelacions ternàries donen lloc a una nova relació que conté tots els atributs que formen les claus primàries de les tres entitats interrelacionades i tots els atributs de la interrelació. La clau primària de la relació depèn de la connectivitat:

- Connectivitat 1:1:1: La relació que s'obté de la seva transformació té com a clau primària els atributs que formen la clau primària de dues entitats qualsevol de les tres interrelacionades.
- Connectivitat 1:1:N: La relació que s'obté de la seva transformació té com a clau primària els atributs que formen la clau primària de l'entitat del costat N i els atributs que formen la clau primària de qualsevol de les dues entitats que estan connectades amb 1.
- Connectivitat 1:N:M: La relació que s'obté de la seva transformació té com a clau primària tots els atributs que formen les claus primàries de les dues entitats interrelacionades dels costats de la interrelació etiquetats amb M i amb N.
- Connectivitat N:M:P: La relació que s'obté de la seva transformació té com a clau primària tots els atributs que formen les claus primàries de les tres entitats interrelacionades.

Les entitats superclasse i subclasse que formen part d'una generalització/especialització es transforma en una relació on:

- La relació de l'entitat superclasse té com a clau primària la clau de l'entitat superclasse i conté tots els atributs comuns.
- Les relacions de les entitats subclasse tenen com a clau primària la clau de l'entitat superclasse i contenen els atributs específics de la subclasse.

4.2.1. Model relacional

CALENDARI (data, diaSetmana, festiu)

ASSIGNATURA (idAssignatura, nom)

CURS (idCurs, anyAcademic, nom, tutor)
{tutor} és clau forana cap a PROFESSOR (DNI)

ASSIGNATURES_CURS (curs, anyAcademic, assignatura)
{curs} és clau forana cap a CURS (idCurs)
{anyAcademic} és clau forana cap a CURS (anyAcademic)
{assignatura} és clau forana cap a ASSIGNATURA (idAssignatura)

CALENDARI_ESCOLAR (data, horaInici, horaFi, curs, anyAcademic, assignatura)
{data} és clau forana cap a CALENDARI (data)
{curs} és clau forana cap a CURS (idCurs)
{anyAcademic} és clau forana cap a CURS (anyAcademic)
{assignatura} és clau forana cap a ASSIGNATURA (idAssignatura)

ALUMNE (DNI, expedient, nom, cognom1, cognom2, dataNaixement, adreça, ciutat, codiPostal, provincia, telefon, mobil, email)
Entitat subclasse de PERSONA

{expedient} és clau alternativa

MATRICULA (alumne, curs, anyAcademic)

{alumne} és clau forana cap a ALUMNE (DNI)

{curs} és clau forana cap a CURS (idCurs)

{anyAcademic} és clau forana cap a CURS (anyAcademic)

PROFESSOR (DNI, nom, cognom1, cognom2, dataNaixement, adreça, ciutat, codiPostal, provincia, telefon, mobil, email)

Entitat subclasse de PERSONA

IMPARTIR (professor, curs, anyAcademic, assignatura)

{professor} és clau forana cap a PROFESSOR (DNI)

{assignatura} és clau forana cap a ASSIGNATURA (idAssignatura)

{curs} és clau forana cap a CURS (idCurs)

{anyAcademic} és clau forana cap a CURS (anyAcademic)

CALENDARI_TUTORIA (data, curs, anyAcademic, horaInici, horaFi, tutor, horariAlumnes)

{data} és clau forana cap a CALENDARI (data)

{curs} és clau forana cap a CURS (idCurs)

{anyAcademic} és clau forana cap a CURS (anyAcademic)

{tutor} és clau forana cap a PROFESSOR (DNI)

REUNIO_ALUMNES (alumne, professor, data, hora, tema)

{alumne} és clau forana cap a ALUMNE (DNI)

{professor} és clau forana cap a PROFESSOR (DNI)

REUNIO_PARES (alumne, professor, data, hora, tema)

{alumne} és clau forana cap a ALUMNE (DNI)

{professor} és clau forana cap a PROFESSOR (DNI)

TIPUS_AMONESTACIO (idTipus, tipus, descripcio)

{tipus} és clau alternativa

TIPUS_SANCIO (idTipus, tipus, activacio, tipusAmonestacio, valor)

{tipusAmonestacio} és clau forana cap a TIPUS_AMONESTACIO (idTipus)

{tipus} és clau alternativa

AMONESTACIO (idAmonestacio, data, hora, comunicacioPares, gravetat, amonestat, amonestador, curs, anyAcademic, assignatura, tipusAmonestacio)

{amonestat} és clau forana cap a ALUMNE (DNI)

{amonestador} és clau forana cap a PROFESSOR (DNI)

{curs} és clau forana cap a CURS (idCurs)

{anyAcademic} és clau forana cap a CURS (anyAcademic)

{assignatura} és clau forana cap a ASSIGNATURA (idAssignatura)

{tipusAmonestacio} és clau forana cap a TIPUS_AMONESTACIO (idTipus)

SANCIO (idSancio, data, motiu, resolucio, sancionat, curs, anyAcademic, tipusSancio)

{sancionat} és clau forana cap a ALUMNE (DNI)

{curs} és clau forana cap a CURS (idCurs)

{anyAcademic} és clau forana cap a CURS (anyAcademic)
{tipusSancio} és clau forana cap a TIPUS_SANCIO (idTipus)

LOG_TAULA (procediment, data, hora, entrada, sortida)

NUMERO_AMONESTACIONS_ALUMNE (DNI, numAmonestacions)

NUMERO_SANCIONS_ALUMNEANY (DNI, anyAcademic, numSancions)

MITJANA_AMONESTACIONS_PROFESSORANY (DNI, anyAcademic, mitjanaAmonestacions, numAmonestacions)

NUMERO_SANCIONS_CURSANY (curso, anyAcademic, numSancions)

ALUMNE_MES_SANCIONAT_ANY (DNI, anyAcademic, numSancions)

PROFESSOR_MES_AMONESTADOR_CURS (DNI, curso, anyAcademic, numAmonestacions)

MITJANA_SANCIONS_CURS (curso, anyAcademic, mitjanaSancions, numSancions)

ALUMNES_SENSE_AMONESTACIONS (DNI)

Canvis realitzats respecte del model conceptual:

- Com que l'entitat PERSONA no s'utilitzarà mai, ja que es tracta d'una generalització/especialització total, es decideix eliminar l'entitat PERSONA del model relacional, quedant només les seves subclasses ALUMNE i PROFESSOR.

4.2.2. Restriccions d'integritat

Les restriccions d'integritat per tractament de claus foranes són:

- No es pot esborrar un tipus d'amonestació si hi ha una amonestació que hi fa referència.
- No es pot esborrar un curs si hi ha una amonestació que hi fa referència.
- No es pot esborrar una assignatura si hi ha una amonestació que hi fa referència.
- No es pot esborrar un professor si hi ha una amonestació que hi fa referència.
- No es pot esborrar un alumne si hi ha una amonestació que hi fa referència.
- No es pot esborrar un tipus de sanció si hi ha una sanció que hi fa referència.
- No es pot esborrar un tipus d'amonestació si hi ha un tipus de sanció que hi fa referència.
- No es pot esborrar un alumne si hi ha una sanció que hi fa referència.
- Si s'esborra un curs, s'esborraran les assignatures i els calendaris associats a aquest curs.
- Si s'esborra una assignatura, s'esborraran els cursos i els calendaris associats a aquesta assignatura.
- Si s'esborra un professor que feia de tutor en un curs, tant l'atribut *tutor* de la relació CURS com de la relació CALENDARI_TUTORIA prendrà valor nul.

4.2.3. Regles de negoci

Les regles de negoci són restriccions que han de complir les operacions sobre les dades.

Les regles de negoci de la base de dades són:

- L'atribut *comunicacioPares* de la relació AMONESTACIO només pot prendre els valors “s” o “n”.
- L'atribut *diaSetmana* de la relació CALENDARI només pot prendre els valors “dilluns”, “dimarts”, “dimecres”, “dijous” o “divendres”.
- L'atribut *festiu* de la relació CALENDARI només pot prendre els valors “s” o “n”.
- L'atribut *horariAlumnes* de la relació CALENDARI_TUTORIA només pot prendre els valors “a” o “p”.
- A les relacions CALENDARI_ESCOLAR i CALENDARI_TUTORIA, l'atribut *horaFi* ha de ser posterior a *horaInici*.
- Quan s'insereixi tant a la relació CALENDARI_ESCOLAR com a la relació CALENDARI_TUTORIA, l'atribut *data* no pot estar marcat com a festiu en la relació CALENDARI.
- Cada sanció ha d'estar lligada a una amonestació.
- Quan l'alumne arribi tard tres vegades, s'activarà una sanció automàtica que obligarà a l'alumne a quedar-se una hora extra d'estudi durant una setmana.
- Quan l'alumne faci molt de soroll a classe cinc vegades, s'activarà una sanció automàtica que obligarà a l'alumne a fer una prova d'avaluació extra.

4.3. Disseny físic

El disseny físic descriu les relacions base amb els seus atributs, claus primàries i foranes i regles d'integritat. També s'implementen els programes d'aplicació per les transaccions i els procediments propis del sistema.

El sistema gestor de bases de dades que es farà servir serà Oracle 10g, per tant, s'ha de convertir el model lògic obtingut a PL/SQL.

4.3.1. Taules

En els següents apartats s'explica en detall la implementació de les taules necessàries per a l'execució de l'aplicació. Tot i que el codi SQL de creació de les taules es mostra a continuació, s'ha decidit no incloure el codi de creació dels disparadors, sobretot els que s'encarreguen d'omplir les taules del mòdul estadístic, degut a la seva longitud. Tot aquest codi es troba en un fitxer a part.

4.3.1.1. Taula CALENDARI

```
CREATE TABLE Calendari (
  data DATE CONSTRAINT PK_Calendari PRIMARY KEY,
  diaSetmana VARCHAR2(9)
  CONSTRAINT NN_DiaSetmana NOT NULL
  CONSTRAINT CH_DiaSetmana CHECK (LOWER(diaSetmana) IN ('dilluns', 'dimarts',
    'dimecres', 'dijous', 'divendres')),
  festiu CHAR(1)
  CONSTRAINT NN_Festiu NOT NULL
  CONSTRAINT CH_Festiu CHECK (LOWER(festiu) IN ('s', 'n'))
);
```

4.3.1.2. Taula ASSIGNATURA

```
CREATE TABLE Assignatura (  
    idAssignatura NUMBER(5) CONSTRAINT PK_Assignatura PRIMARY KEY,  
    nom VARCHAR2(20) CONSTRAINT NN_NomAssignatura NOT NULL  
);
```

Creació de la seqüència per a generar els números de la clau primària:

```
CREATE SEQUENCE seq_idAssignatura  
    START WITH 1  
    INCREMENT BY 1  
    NOCACHE  
    NOCYCLE;
```

Creació del disparador per a assignar el número generat per la seqüència *seq_idAssignatura* a *idAssignatura* cada vegada que s'insereixi una assignatura:

```
CREATE OR REPLACE  
TRIGGER inserir_Assignatura  
    BEFORE INSERT ON Assignatura  
    FOR EACH ROW  
BEGIN  
    SELECT seq_idAssignatura.NEXTVAL INTO :NEW.idAssignatura  
    FROM DUAL;  
END inserir_Assignatura;
```

4.3.1.3. Taula CURS

```
CREATE TABLE Curs (  
    idCurs NUMBER(5),  
    anyAcademic NUMBER(4),  
    nom VARCHAR2(20) CONSTRAINT NN_NomCurs NOT NULL,  
    tutor,  
    CONSTRAINT PK_Curs PRIMARY KEY (IdCurs, anyAcademic),  
    CONSTRAINT FK_Curs FOREIGN KEY (tutor) REFERENCES Professor (DNI) ON DELETE SET  
    NULL  
);
```

ON DELETE SET NULL indica que: si s'esborra un professor que feia de tutor d'un curs, l'atribut tutor prendrà valor null.

Creació de la seqüència per a generar els números de la clau primària *idCurs* de la taula Curs:

```
CREATE SEQUENCE seq_idCurs  
    START WITH 1  
    INCREMENT BY 1  
    NOCACHE  
    NOCYCLE;
```

Creació del disparador per a assignar el número generat per la seqüència *seq_idCurs* a *idCurs* cada vegada que s'insereixi un curs:

```
CREATE OR REPLACE  
TRIGGER inserir_Curs  
    BEFORE INSERT ON Curs  
    FOR EACH ROW  
BEGIN  
    SELECT seq_idCurs.NEXTVAL INTO :NEW.idCurs  
    FROM DUAL;  
END inserir_Curs;
```

Cada vegada que s'insereixi un nou curs, s'executarà un disparador que posarà dades en una de les taules que formen el mòdul estadístic, en concret aquest disparador:

- Inserirà una fila a la taula *Numero_Sancions_CursAny* inicialitzant el valor de l'atribut *numSancions*.

Com s'ha dit abans, el codi de creació d'aquest disparador es troba en un fitxer apart.

4.3.1.4. Taula ASSIGNATURES_CURS

```
CREATE TABLE Assignatures_Curs (
  curs,
  anyAcademic,
  assignatura,
  CONSTRAINT PK_Assignatures_Curs PRIMARY KEY (curs, anyAcademic, assignatura),
  CONSTRAINT FK1_Assignatures_Curs FOREIGN KEY (curs, anyAcademic) REFERENCES Curs
    (idCurs, anyAcademic) ON DELETE CASCADE,
  CONSTRAINT FK2_Assignatures_Curs FOREIGN KEY (assignatura) REFERENCES Assignatura
    (idAssignatura)
);
```

ON DELETE CASCADE indica que:

- Si s'esborra un curs s'eliminaran els registres d'aquesta taula associats a aquest curs.

4.3.1.5. Taula CALENDARI_ESCOLAR

```
CREATE TABLE Calendari_Escolar (
  data DATE,
  horaInici CHAR(5)
  CONSTRAINT CH_HoraIniciCalEsc CHECK (REGEXP_LIKE (horaInici,
    '([[:digit:]]{2})\:([[:digit:]]{2})'),
  horaFi CHAR(5)
  CONSTRAINT CH_HoraFiCalEsc CHECK (REGEXP_LIKE (horaFi,
    '([[:digit:]]{2})\:([[:digit:]]{2})'),
  curs NUMBER(5),
  anyAcademic NUMBER(4),
  assignatura NUMBER(5) CONSTRAINT NN_Assignatura NOT NULL,
  CONSTRAINT PK_Calendari_Escolar PRIMARY KEY (data, horaInici, horaFi, curs,
    anyAcademic),
  CONSTRAINT FK1_Calendari_Escolar FOREIGN KEY (data) REFERENCES Calendari (data),
  CONSTRAINT FK2_Calendari_Escolar FOREIGN KEY (curs, anyAcademic, assignatura)
    REFERENCES Assignatures_Curs (curs, anyAcademic, assignatura) ON DELETE
    CASCADE,
  CONSTRAINT CH_HoresCalEsc CHECK (horaFi > horaInici)
);
```

ON DELETE CASCADE indica que:

- Si s'esborra una assignatura d'un curs, s'eliminaran els calendaris associats a aquesta relació assignatura-curs.

Tot i que en els model conceptual i lògic s'havia decidit escollir tres claus foranes: una per la data de la taula Calendari, una altre per el curs i l'any acadèmic de la taula Curs i una altra per l'assignatura. Per implementar el calendari escolar, s'ha decidit eliminar les dues ultimes claus foranes i fer-ne una formada per el curs, l'any acadèmic i l'assignatura de la taula Assignatures_Curs, d'aquesta manera ens assegurem que l'assignatura del calendari pertany realment el curs indicat.

Creació del disparador que s'executarà abans que s'insereixi un nou calendari escolar que comprovarà que el dia introduït no sigui festiu i que les hores d'inici i fi estiguin dintre de l'interval [00:00 – 23:59]:

```
CREATE OR REPLACE
TRIGGER INSERIR_CALENDARI_ESCOLAR
BEFORE INSERT ON CALENDARI_ESCOLAR
FOR EACH ROW
DECLARE
    t_festiu CHAR(1);
BEGIN
    -- Comprova que la data introduïda no caigui en dia festiu
    SELECT festiu INTO t_festiu FROM Calendari WHERE data = :NEW.data;
    IF (LOWER(t_festiu) = 's') THEN
        raise_application_error (-20001, 'ERROR: La data no és correcta, el dia
        introduït és festiu.');
```

4.3.1.6. Taula ALUMNE

```
CREATE TABLE Alumne (
    DNI VARCHAR2(10) CONSTRAINT PK_Alumne PRIMARY KEY,
    expedient VARCHAR2(8) CONSTRAINT NN_Expedient NOT NULL,
    nom VARCHAR2(30) CONSTRAINT NN_NomAlumne NOT NULL,
    cognom1 VARCHAR2(40) CONSTRAINT NN_Cognom1Alumne NOT NULL,
    cognom2 VARCHAR2(40) CONSTRAINT NN_Cognom2Alumne NOT NULL,
    dataNaixement DATE CONSTRAINT NN_DataNaixementAlumne NOT NULL,
    adreça VARCHAR2(50) CONSTRAINT NN_AdreçaAlumne NOT NULL,
    ciutat VARCHAR2(30) CONSTRAINT NN_CiutatAlumne NOT NULL,
    codiPostal NUMBER(5) CONSTRAINT NN_CodiPostalAlumne NOT NULL,
    provincia VARCHAR2(30) CONSTRAINT NN_ProvinciaAlumne NOT NULL,
    telefon NUMBER(9),
    mobil NUMBER(9),
    email VARCHAR2(20),
    CONSTRAINT UN_Alumne UNIQUE (expedient)
);
```

Cada vegada que s'insereixi un nou alumne, s'executarà un disparador que posarà dades en varies de les taules que formen el mòdul estadístic, en concret aquest disparador:

- Inserirà una fila a la taula *Numero_Amonestacions_Alumne* inicialitzant el valor de l'atribut *numAmonestacions*.
- Inserirà una fila a la taula *Alumnes_Sense_Amonestacions*.

Com s'ha dit abans, el codi de creació d'aquest disparador es troba en un fitxer apart.

4.3.1.7. Taula MATRICULA

```
CREATE TABLE Matricula (
```

```

alumne VARCHAR2(10),
curs NUMBER(5),
anyAcademic NUMBER(4),
CONSTRAINT PK_Matricula PRIMARY KEY (alumne, curs, anyAcademic),
CONSTRAINT FK1_Matricula FOREIGN KEY (alumne) REFERENCES Alumne (DNI),
CONSTRAINT FK2_Matricula FOREIGN KEY (curs, anyAcademic) REFERENCES Curs (idCurs,
anyAcademic) ON DELETE CASCADE
);

```

ON DELETE CASCADE indica que:

- Si s'esborra un curs, s'eliminaran les matricules d'aquest curs.

Cada vegada que s'insereixi un nova matricula, s'executarà un disparador que posarà dades en varies de les taules que formen el mòdul estadístic, en concret aquest disparador:

- Inserirà una fila a la taula *Numero_Sancions_AlumneAny* inicialitzant el valor de l'atribut *numSancions*.
- Inserirà una fila a la taula *Mitjana_Sancions_Curs* inicialitzant el valor de l'atribut *mitjanaSancions* i *numSancions*.

Com s'ha dit abans, el codi de creació d'aquest disparador es troba en un fitxer apart.

4.3.1.8. Taula PROFESSOR

```

CREATE TABLE Professor (
DNI VARCHAR2(10) CONSTRAINT PK_Professor PRIMARY KEY,
nom VARCHAR2(30) CONSTRAINT NN_NomProfessor NOT NULL,
cognom1 VARCHAR2(40) CONSTRAINT NN_Cognom1Professor NOT NULL,
cognom2 VARCHAR2(40) CONSTRAINT NN_Cognom2Professor NOT NULL,
dataNaixement DATE CONSTRAINT NN_DataNaixementProfessor NOT NULL,
adreça VARCHAR2(50) CONSTRAINT NN_AdreçaProfessor NOT NULL,
ciutat VARCHAR2(30) CONSTRAINT NN_CiutatProfessor NOT NULL,
codiPostal NUMBER(5) CONSTRAINT NN_CodiPostalProfessor NOT NULL,
provincia VARCHAR2(30) CONSTRAINT NN_ProvinciaProfessor NOT NULL,
telefon NUMBER(9),
mobil NUMBER(9),
email VARCHAR2(20)
);

```

4.3.1.9. Taula IMPARTIR

```

CREATE TABLE Impartir (
professor VARCHAR2(10),
assignatura NUMBER(5),
curs NUMBER(5),
anyAcademic NUMBER(4),
CONSTRAINT PK_Impartir PRIMARY KEY (professor, assignatura, curs, anyAcademic),
CONSTRAINT FK1_Impartir FOREIGN KEY (professor) REFERENCES Professor (DNI),
CONSTRAINT FK2_Impartir FOREIGN KEY (assignatura, curs, anyAcademic) REFERENCES
Assignatures_Curs (assignatura, curs, anyAcademic) ON DELETE CASCADE
);

```

ON DELETE CASCADE indica que:

- Si s'esborra una assignatura d'un curs, s'eliminaran els registres d'aquesta taula associats a aquesta relació assignatura-curs.

Tot i que en els model conceptual i lògic s'havia decidit escollir tres claus foranes: una per el professor, una altre per el curs i l'any acadèmic de la taula Curs i una altra per l'assignatura. Per implementar la taula Impartir s'ha decidit eliminar les dues ultimes claus foranes i fer-ne una formada per el curs, l'any acadèmic i l'assignatura de la taula Assigatures_Curs, d'aquesta manera ens assegurem que l'assignatura pertany realment el curs indicat.

Cada vegada que s'insereixi un nova fila a la taula Impartir, s'executarà un disparador que posarà dades en una de les taules que formen el mòdul estadístic, en concret aquest disparador:

- Inserirà una fila a la taula *Mitjana_Amonestacions_ProfessorAny* inicialitzant el valor de l'atribut *mitjanaAmonestacions* i *numAmonestacions*.

Com s'ha dit abans, el codi de creació d'aquest disparador es troba en un fitxer apart.

4.3.1.10. Taula CALENDARI_TUTORIA

```
CREATE TABLE Calendari_Tutoria (
  data DATE,
  horaInici CHAR(5)
    CONSTRAINT CH_HoraIniciCalTut CHECK (REGEXP_LIKE (horaInici,
      '([[:digit:]]{2})\:([[:digit:]]{2})')),
  horaFi CHAR(5)
    CONSTRAINT CH_HoraFiCalTut CHECK (REGEXP_LIKE (horaFi,
      '([[:digit:]]{2})\:([[:digit:]]{2})')),
  curs NUMBER(5),
  anyAcademic NUMBER(4),
  tutor VARCHAR2(10),
  horariAlumnes CHAR(1)
    CONSTRAINT NN_horariAlumnes NOT NULL
    CONSTRAINT CH_HorariAlumnes CHECK (LOWER(horariAlumnes) IN ('a', 'p')),
  CONSTRAINT PK_Calendari_Tutoria PRIMARY KEY (data, horaInici, horaFi, curs,
    anyAcademic),
  CONSTRAINT FK1_Calendari_Tutoria FOREIGN KEY (data) REFERENCES Calendari (data),
  CONSTRAINT FK2_Calendari_Tutoria FOREIGN KEY (curs, anyAcademic) REFERENCES Curs
    (idCurs, anyAcademic),
  CONSTRAINT FK3_Calendari_Tutoria FOREIGN KEY (tutor) REFERENCES Professor (DNI)
    ON DELETE SET NULL,
  CONSTRAINT CH_HoresCalTut CHECK (horaFi > horaInici)
);
```

ON DELETE SET NULL indica que: si s'esborra un professor que feia de tutor d'un curs, l'atribut tutor prendrà valor null.

Com amb el calendari escolar, es crea un disparador que s'executarà abans que s'insereixi un nou calendari de tutoria que comprovarà que el dia introduït no sigui festiu i que les hores d'inici i fi estiguin dintre de l'interval [00:00 – 23:59]. Com que és molt semblant el del calendari escolar, no el mostrarem aquí.

4.3.1.11. Taula REUNIO_ALUMNE

```
CREATE TABLE Reunio_Alumne (
  alumne VARCHAR2(10),
  professor VARCHAR2(10),
  dataiHora DATE,
  tema VARCHAR2(50),
```

```

CONSTRAINT PK_Reunio_Alumne PRIMARY KEY (alumne, professor, dataiHora),
CONSTRAINT FK1_Reunio_Alumne FOREIGN KEY (alumne) REFERENCES Alumne (DNI),
CONSTRAINT FK2_Reunio_Alumne FOREIGN KEY (professor) REFERENCES Professor (DNI)
);

```

4.3.1.12. Taula REUNIO_PARES

```

CREATE TABLE Reunio_Pares (
  alumne VARCHAR2(10),
  professor VARCHAR2(10),
  dataiHora DATE,
  tema VARCHAR2(50),
  CONSTRAINT PK_Reunio_Pares PRIMARY KEY (alumne, professor, dataiHora),
  CONSTRAINT FK1_Reunio_Pares FOREIGN KEY (alumne) REFERENCES Alumne (DNI),
  CONSTRAINT FK2_Reunio_Pares FOREIGN KEY (professor) REFERENCES Professor (DNI)
);

```

4.3.1.13. Taula TIPUS_AMONESTACIO

```

CREATE TABLE Tipus_Amonestacio (
  idTipus NUMBER(5) CONSTRAINT PK_Tipus_Amonestacio PRIMARY KEY,
  tipus VARCHAR2(30) CONSTRAINT NN_TipusAmonestacio NOT NULL,
  descripcio VARCHAR2(50) CONSTRAINT NN_Descripcio NOT NULL,
  CONSTRAINT UN_Tipus_Amonestacio UNIQUE (tipus)
);

```

Creació de la seqüència per a generar els números de la clau primària *idTipus*:

```

CREATE SEQUENCE seq_idTipus_Amonestacio
  START WITH 1
  INCREMENT BY 1
  NOCACHE
  NOCYCLE;

```

Creació del disparador per a assignar el número generat per la seqüència *seq_idTipus_Amonestacio* a *idTipus* cada vegada que s'insereixi un tipus d'amonestació:

```

CREATE OR REPLACE
TRIGGER inserir_Tipus_Amonestacio
  BEFORE INSERT ON Tipus_Amonestacio
  FOR EACH ROW
BEGIN
  SELECT seq_idTipus_Amonestacio.NEXTVAL INTO :NEW.idTipus
  FROM DUAL;
END inserir_Tipus_Amonestacio;

```

Inserció dels tipus d'amonestacions establertes pels professors inicialment:

```

INSERT INTO Tipus_Amonestacio (tipus, descripcio) VALUES ('alumne_arriba_tard',
  'Quan l'alumne arriba tard a classe.');
```

```

INSERT INTO Tipus_Amonestacio (tipus, descripcio) VALUES ('alumne_soroll',
  'L'alumne fa molt de soroll a classe.');
```

```

INSERT INTO Tipus_Amonestacio (tipus, descripcio) VALUES ('alumne_malparlat',
  'L'alumne és malparlat.');
```

```

INSERT INTO Tipus_Amonestacio (tipus, descripcio) VALUES ('alumne_no_deures',
  'L'alumne no fa els deures.');
```

4.3.1.14. Taula TIPUS_SANCIO

```

CREATE TABLE Tipus_Sancio (
  idTipus NUMBER(5) CONSTRAINT PK_Tipus_Sancio PRIMARY KEY,
  tipus VARCHAR2(60) CONSTRAINT NN_TipusSancio NOT NULL,
  activacio VARCHAR2(60) CONSTRAINT NN_Activacio NOT NULL,
  tipusAmonestacio NUMBER(5) CONSTRAINT NN_TipAmonestacio NOT NULL,
  valor NUMBER(2) CONSTRAINT NN_Valor NOT NULL,
  CONSTRAINT FK_Tipus_Sancio FOREIGN KEY (tipusAmonestacio) REFERENCES
    Tipus_Amonestacio (idTipus),
  CONSTRAINT UN_Tipus_Sancio UNIQUE (tipus)
);

```

Per defecte, no es pot esborrar un tipus d'amonestació si hi ha un tipus de sanció que hi fa referència.

Creació de la seqüència per a generar els números de la clau primària *idTipus*:

```

CREATE SEQUENCE seq_idTipus_Sancio
  START WITH 1
  INCREMENT BY 1
  NOCACHE
  NOCYCLE;

```

Creació del disparador per a assignar el número generat per la seqüència *seq_idTipus_Sancio* a *idTipus* cada vegada que s'insereixi un tipus de sanció:

```

CREATE OR REPLACE
  TRIGGER inserir_Tipus_Sancio
  BEFORE INSERT ON Tipus_Sancio
  FOR EACH ROW
  BEGIN
    SELECT seq_idTipus_Sancio.NEXTVAL INTO :NEW.idTipus
    FROM DUAL;
  END inserir_Tipus_Sancio;

```

Inserció dels tipus de sancions establertes pels professors inicialment:

```

INSERT INTO Tipus_Sancio (tipus, activacio, tipusAmonestacio, valor) VALUES
  ('Quedar-se una hora extra d'estudi durant una setmana', 'Automàticament
  quan l'alumne arriba tard tres vegades.', 1, 3);
INSERT INTO Tipus_Sancio (tipus, activacio, tipusAmonestacio, valor) VALUES ('Prova
  d'avaluació extra', 'Automàticament quan fan molt de soroll a classe 5
  vegades.', 2, 5);
INSERT INTO Tipus_Sancio (tipus, activacio, tipusAmonestacio, valor) VALUES
  ('Sanció personalitzada', 'Introducció manual pel professor.', 3, 0);

```

4.3.1.15. Taula AMONESTACIO

```

CREATE TABLE Amonestacio (
  idAmonestacio NUMBER(5) CONSTRAINT PK_Amonestacio PRIMARY KEY,
  dataiHora DATE CONSTRAINT NN_DataiHora NOT NULL,
  comunicacioPares CHAR(1)
  CONSTRAINT NN_ComunicacioPares NOT NULL
  CONSTRAINT CH_ComunicacioPares CHECK (LOWER(comunicacioPares) IN ('s', 'n')),
  gravetat VARCHAR2(10) CONSTRAINT NN_Gravetat NOT NULL,
  amonestat VARCHAR2(10) CONSTRAINT NN_Amonestat NOT NULL,
  amonestador VARCHAR2(10) CONSTRAINT NN_Amonestador NOT NULL,
  curs NUMBER(5) CONSTRAINT NN_CursAmonestacio NOT NULL,
  anyAcademic NUMBER(4) CONSTRAINT NN_AnyAcademicAmonestacio NOT NULL,
  assignatura NUMBER(5) CONSTRAINT NN_AssignaturaAmonestacio NOT NULL,
  tipusAmonestacio NUMBER(5) CONSTRAINT NN_TAmonestacio NOT NULL,
  CONSTRAINT FK1_Amonestacio FOREIGN KEY (amonestat) REFERENCES Alumne (DNI),
  CONSTRAINT FK2_Amonestacio FOREIGN KEY (amonestador) REFERENCES Professor (DNI),
  CONSTRAINT FK3_Amonestacio FOREIGN KEY (curs, anyAcademic, assignatura)
  REFERENCES Assignatures_Curs (curs, anyAcademic, assignatura),
  CONSTRAINT FK4_Amonestacio FOREIGN KEY (tipusAmonestacio) REFERENCES
    Tipus_Amonestacio (idTipus)
);

```



```
);
```

Per defecte, no es pot esborrar un alumne, un professor, un curs, una assignatura o un tipus d'amonestació, si hi ha una amonestació que els fa referència.

Creació de la seqüència per a generar els números de la clau primària *idAmonestacio*:

```
CREATE SEQUENCE seq_idAmonestacio
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

Creació del disparador per a assignar el número generat per la seqüència *seq_idAmonestacio* a *idAmonestacio* cada vegada que s'insereixi un tipus de amonestació:

```
CREATE OR REPLACE
TRIGGER inserir_Amonestacio
BEFORE INSERT ON Amonestacio
FOR EACH ROW
BEGIN
SELECT seq_idAmonestacio.NEXTVAL INTO :NEW.idAmonestacio
FROM DUAL;
END inserir_Amonestacio;
```

Creació del disparador que s'executarà, cada vegada, després d'inserir un nova amonestació per a activar una sanció automàtica, si és el cas:

```
CREATE OR REPLACE
TRIGGER ACTIVAR_SANCIO
AFTER INSERT ON AMONESTACIO
FOR EACH ROW
DECLARE
t_numAmonestacions NUMBER(3) := 0;
t_idTipus NUMBER(5);
t_valor NUMBER(2) := 0;
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
SELECT COUNT(*) INTO t_numAmonestacions
FROM Amonestacio
WHERE amonestat = :NEW.amonestat AND tipusAmonestacio = :NEW.tipusAmonestacio;
SELECT idTipus, valor INTO t_idTipus, t_valor
FROM Tipus_Sancio
WHERE tipusAmonestacio = :NEW.tipusAmonestacio;
IF (t_numAmonestacions >= t_valor) THEN
INSERT INTO Sancio (data, motiu, resolucio, sancionat, curs, anyAcademic,
tipusSancio) VALUES (SYSDATE, 'Sanció automàtica', 'Sanció automàtica',
:NEW.amonestat, :NEW.curs, :NEW.anyAcademic, t_idTipus);
END IF;
COMMIT;
END activar_Sancio;
```

Cada vegada que s'insereixi una nova amonestació, s'executarà un disparador que posarà dades en una de les taules que formen el mòdul estadístic, en concret aquest disparador:

- Actualitzarà el valor de l'atribut *numAmonestacions* de la taula *Numero_Amonestacions_Alumne*.
- Actualitzarà el valor de l'atribut *mitjanaAmonestacions* i *numAmonestacions* de la taula *Mitjana_Amonestacions_ProfessorAny*.
- Actualitzarà el valor dels atributs *numAmonestacions* i *DNI* del professor més amonestador de la taula *Professor_Mes_Amonestador_Curs*.
- Eliminarà la fila de l'alumne amonestat de la taula *Alumnes_Sense_Amonestacions*.

Com s'ha dit abans, el codi de creació d'aquest disparador es troba en un fitxer apart.

4.3.1.16. Taula SANCIO

```
CREATE TABLE Sancio (
  idSancio NUMBER(5) CONSTRAINT PK_Sancio PRIMARY KEY,
  data DATE CONSTRAINT NN_DataSancio NOT NULL,
  motiu VARCHAR2(50) CONSTRAINT NN_Motiu NOT NULL,
  resolucio VARCHAR2(50) CONSTRAINT NN_Resolucio NOT NULL,
  sancionat VARCHAR2(10) CONSTRAINT NN_Sancionat NOT NULL,
  curs NUMBER(5) CONSTRAINT NN_CursSancio NOT NULL,
  anyAcademic NUMBER(4) CONSTRAINT NN_AnyAcademicSancio NOT NULL,
  tipusSancio NUMBER(5) CONSTRAINT NN_TipSancio NOT NULL,
  CONSTRAINT FK1_Sancio FOREIGN KEY (sancionat) REFERENCES Alumne (DNI),
  CONSTRAINT FK2_Sancio FOREIGN KEY (curs, anyAcademic) REFERENCES Curs (idCurs,
    anyAcademic),
  CONSTRAINT FK3_Sancio FOREIGN KEY (tipusSancio) REFERENCES Tipus_Sancio
    (idTipus)
);
```

Per defecte, no es pot esborrar un alumne o un tipus de sanció, si hi ha una sanció que els fa referència.

Creació de la seqüència per a generar els números de la clau primària *idSancio*:

```
CREATE SEQUENCE seq_idSancio
  START WITH 1
  INCREMENT BY 1
  NOCACHE
  NOCYCLE;
```

Creació del disparador per a assignar el número generat per la seqüència *seq_idSancio* a *idSancio* cada vegada que s'insereixi una sanció:

```
CREATE OR REPLACE
  TRIGGER inserir_Sancio
  BEFORE INSERT ON Sancio
  FOR EACH ROW
  BEGIN
    SELECT seq_idSancio.NEXTVAL INTO :NEW.idSancio
    FROM DUAL;
  END inserir_Sancio;
```

Cada vegada que s'insereixi una nova sanció, s'executarà un disparador que posarà dades en una de les taules que formen el mòdul estadístic, en concret aquest disparador:

- Actualitzarà el valor de l'atribut *numSancions* de la taula *Numero_Sancions_AlumneAny*.
- Actualitzarà el valor de l'atribut *numSancions* de la taula *Numero_Sancions_CursAny*.
- Actualitzarà el valor dels atributs *numSancions* i *DNI* de l'alumne més sancionat de la taula *Alumne_Mes_Sancionat_Any*.
- Actualitzarà el valor de l'atribut *mitjanaSancions* de la taula *Mitjana_Sancions_Curs*.

Com s'ha dit abans, el codi de creació d'aquest disparador es troba en un fitxer apart.

4.3.1.17. Taula LOG_TAULA

```
CREATE TABLE Log_Taula (
  procediment VARCHAR2(30),
  dataiHora DATE CONSTRAINT NN_Data NOT NULL,
  entrada VARCHAR2(512) CONSTRAINT NN_Entrada NOT NULL,
  sortida VARCHAR2(512) CONSTRAINT NN_Sortida NOT NULL,
  CONSTRAINT PK_Log_Taula PRIMARY KEY (procediment, dataiHora)
```

```
);
```

4.3.1.18. Taula NUMERO_AMONESTACIONS_ALUMNE

```
CREATE TABLE Numero_Amonestacions_Alumne (  
  DNI VARCHAR2(10) CONSTRAINT PK_Numero_Amonestacions_Alumne PRIMARY KEY,  
  numAmonestacions NUMBER(3) CONSTRAINT NN_NumAmonestacionsAlumne NOT NULL  
);
```

4.3.1.19. Taula NUMERO_SANCIONS_ALUMNEANY

```
CREATE TABLE Numero_Sancions_AlumneAny (  
  DNI VARCHAR2(10),  
  anyAcademic NUMBER(4),  
  numSancions NUMBER(3) CONSTRAINT NN_NumSancions NOT NULL,  
  CONSTRAINT PK_Numero_Sancions_AlumneAny PRIMARY KEY (DNI, anyAcademic)  
);
```

4.3.1.20. Taula MITJANA_AMONESTACIONS_PROFESSORANY

```
CREATE TABLE Mitjana_Amonestacions_ProfAny (  
  DNI VARCHAR2(10),  
  anyAcademic NUMBER(4),  
  MitjanaAmonestacions NUMBER(3) CONSTRAINT NN_MitjanaAmonestacions NOT NULL,  
  NumAmonestacions NUMBER(3) CONSTRAINT NN_NumAmonestacionsProfAny NOT NULL,  
  CONSTRAINT PK_Mitjana_Amonestany PRIMARY KEY (DNI, anyAcademic)  
);
```

4.3.1.21. Taula NUMERO_SANCIONS_CURSANY

```
CREATE TABLE Numero_Sancions_CursAny (  
  curs NUMBER(5),  
  anyAcademic NUMBER(4),  
  numSancions NUMBER(3) CONSTRAINT NN_NumSancionsCursany NOT NULL,  
  CONSTRAINT PK_Numero_Sancions_CursAny PRIMARY KEY (curs, anyAcademic)  
);
```

4.3.1.22. Taula ALUMNE_MES_SANCIONAT_ANY

```
CREATE TABLE Alumne_Mes_Sancionat_Any (  
  DNI VARCHAR2(10),  
  anyAcademic NUMBER(4),  
  numSancions NUMBER(3) CONSTRAINT NN_NumSancionsMesSancionat NOT NULL,  
  CONSTRAINT PK_Alumne_Mes_Sancionat_Any PRIMARY KEY (DNI, anyAcademic)  
);
```

4.3.1.23. Taula PROFESSOR_MES_AMONESTADOR_CURS

```
CREATE TABLE Professor_Mes_Amonestador_Curs (  
  DNI VARCHAR2(10),  
  curs NUMBER(5),  
  anyAcademic NUMBER(4),
```

```

numAmonestacions NUMBER(3) CONSTRAINT NN_NumAmMesAmonestador NOT NULL,
CONSTRAINT PK_ProfessorMesAmonestadorCurs PRIMARY KEY (DNI, curs, anyAcademic)
);

```

4.3.1.24. Taula MITJANA_SANCIONS_CURS

```

CREATE TABLE Mitjana_Sancions_Curs (
  curs NUMBER(5),
  anyAcademic NUMBER(4),
  MitjanaSancions NUMBER(3) CONSTRAINT NN_MitjanaSancions NOT NULL,
  NumSancions NUMBER(3) CONSTRAINT NN_NumSancionsCurs NOT NULL,
  CONSTRAINT PK_Mitjana_Sancions_Curs PRIMARY KEY (curs, anyAcademic)
);

```

4.3.1.25. Taula ALUMNES_SENSE_AMONESTACIONS

```

CREATE TABLE Alumnes_Sense_Amonestacions (
  DNI VARCHAR2(10) CONSTRAINT PK_Alumnes_Sense_Amonestacio PRIMARY KEY
);

```

4.3.2. Procediments emmagatzemats

En els següents apartats s'explica en detall la implementació dels procediments emmagatzemats necessaris per a l'execució de l'aplicació, la seva implementació es troba en un fitxer a part.

4.3.2.1. Alta calendari

Aquest procediment insereix un nou dia a la taula *Calendari*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Calendari*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pDiaSetmana* i *pFestiu*.
- Cap dels paràmetres pot ser nul.
- El paràmetre *pDiaSetmana* només pot prendre els valors “dilluns”, “dimarts”, “dimecres”, “dijous” o “divendres”.
- El paràmetre *pFestiu* només pot prendre els valors “s” o “n”.
- El dia no existeix a la taula *Calendari*.

Postcondicions:

- Afegeix un nou dia a la taula *Calendari*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Dia inserit.'
 - Si l'execució finalitza amb un error: 'ERROR: ' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.2. Baixa calendari

Aquest procediment elimina un dia de la taula *Calendari*: agafa els paràmetres d'entrada i esborra totes les dades d'una data determinada amb una sentència DELETE a la taula *Calendari*.

Precondicions:

- El paràmetre d'entrada és *pData*.
- El paràmetre no pot ser nul.
- El dia existeix a la taula *Calendari*.

Postcondicions:

- Esborra el dia de la taula *Calendari*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Dia esborrat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.3. Modificació calendari

Aquest procediment modifica un dia de la taula *Calendari*: agafa els paràmetres d'entrada i modifica totes les dades d'una data determinada amb una sentència UPDATE a la taula *Calendari*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pDiaSetmana* i *pFestiu*.
- Cap dels paràmetres pot ser nul.
- El paràmetre *pDiaSetmana* només pot prendre els valors "dilluns", "dimarts", "dimecres", "dijous" o "divendres".
- El paràmetre *pFestiu* només pot prendre els valors "s" o "n".
- La data existeix a la taula *Calendari*.

Postcondicions:

- Modifica el dia de la setmana i el festiu de la data.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Dia modificat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.4. Alta assignatura

Aquest procediment insereix una nova assignatura a la taula *Assignatura*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Assignatura*.

Precondicions:

- El paràmetres d'entrada és *pNom*.
- El paràmetre no pot ser nul.
- L'assignatura no existeix a la taula *Assignatura*.

Postcondicions:

- Afegeix una nova assignatura a la taula *Assignatura*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Assignatura inserida.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.5. Baixa assignatura

Aquest procediment elimina una assignatura de la taula *Assignatura*: agafa els paràmetres d'entrada i esborra totes les dades d'una assignatura determinada amb una sentència DELETE a la taula *Assignatura*.

Precondicions:

- El paràmetre d'entrada és *pAssignatura*.
- El paràmetre no pot ser nul.
- L'assignatura existeix a la taula *Assignatura*.

Postcondicions:

- Esborra l'assignatura de la taula *Assignatura*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Assignatura esborrada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.6. Modificació assignatura

Aquest procediment modifica una assignatura de la taula *Assignatura*: agafa els paràmetres d'entrada i modifica totes les dades d'una assignatura determinada amb una sentència UPDATE a la taula *Assignatura*.

Precondicions:

- Els paràmetres d'entrada són *pIdAssignatura* i *pNom*.
- Cap dels paràmetres pot ser nul.
- L'assignatura existeix a la taula *Assignatura*.

Postcondicions:

- Modifica el nom de l'assignatura.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Assignatura modificada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.7. Alta curs

Aquest procediment insereix un nou curs a la taula *Curs*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Curs*.

Precondicions:

- Els paràmetres d'entrada són *pAnyAcademic*, *pNom* i *pTutor*.
- El paràmetre *pTutor* pot prendre valor nul.
- El tutor existeix a la taula *Professor*.
- El curs no existeix a la taula *Curs*.

Postcondicions:

- Afegeix un nou curs a la taula *Curs*.
- El paràmetre de sortida *RSP* tindrà el valor:

- Si l'execució finalitza amb èxit: 'OK: Curs inserit.'
- Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.8. Baixa curs

Aquest procediment elimina un curs de la taula *Curs*: agafa els paràmetres d'entrada i esborra totes les dades d'un curs determinat amb una sentència DELETE a la taula *Curs*.

Precondicions:

- Els paràmetres d'entrada són *pCurs* i *pAnyAcademic*.
- Cap dels paràmetres poden ser nuls.
- El curs existeix a la taula *Curs*.

Postcondicions:

- Esborra el curs de la taula *Curs*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Curs esborrat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.9. Modificació curs

Aquest procediment modifica un curs de la taula *Curs*: agafa els paràmetres d'entrada i modifica totes les dades d'un curs i any acadèmic determinat amb una sentència UPDATE a la taula *Curs*.

Precondicions:

- Els paràmetres d'entrada són *pIdCurs*, *pAnyAcademic*, *pNom* i *pTutor*.
- Cap dels paràmetres pot ser nul.
- El tutor existeix a la taula *Professor*.
- El curs existeix a la taula *Curs*.

Postcondicions:

- Modifica el nom i el tutor del curs.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Curs modificat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.10. Alta assignatures curs

Aquest procediment insereix una nova associació curs-assignatura a la taula *Assignatures_Curs*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Assignatures_Curs*.

Precondicions:

- Els paràmetres d'entrada són *pCurs*, *pAnyAcademic* i *pAssignatura*.

- Cap dels paràmetres pot ser nul.
- El curs i any acadèmic existeixen a la taula *Curs*.
- L'assignatura existeix a la taula *Assignatura*.
- L'associació curs-assignatura no existeix a la taula *Assignatures_Curs*.

Postcondicions:

- Afegeix una nova associació curs-assignatura a la taula *Assignatures_Curs*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Associació curs-assignatura inserida.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.11. Alta calendari escolar

Aquest procediment insereix un nou calendari escolar a la taula *Calendari_Escolar*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Calendari_Escolar*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pHoraInici*, *pHoraFi*, *pCurs*, *pAnyAcademic* i *pAssignatura*.
- Cap dels paràmetres pot ser nul.
- El paràmetre *pHoraFi* ha de ser posterior al paràmetre *pHoraInici*.
- La data existeix a la taula *Calendari*.
- El curs i any acadèmic existeixen a la taula *Curs*.
- L'assignatura existeix a la taula *Assignatura*.
- El calendari escolar no existeix a la taula *Calendari_Escolar*.

Postcondicions:

- Afegeix un nou calendari a la taula *Calendari_Escolar*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Calendari escolar inserit.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.12. Baixa calendari escolar

Aquest procediment elimina un calendari escolar de la taula *Calendari_Escolar*: agafa els paràmetres d'entrada i esborra totes les dades d'un calendari determinat amb una sentència DELETE a la taula *Calendari_Escolar*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pHoraInici*, *pHoraFi*, *pCurs* i *pAnyAcademic*.
- Cap dels paràmetres pot ser nul.
- El calendari existeix a la taula *Calendari_Escolar*.

Postcondicions:

- Esborra el calendari escolar de la taula *Calendari_Escolar*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Calendari escolar esborrat.'

- Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.13. Modificació calendari escolar

Aquest procediment modifica un calendari escolar de la taula *Calendari_Escolar*: agafa els paràmetres d'entrada i modifica totes les dades d'un calendari determinat amb una sentència UPDATE a la taula *Calendari_Escolar*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pHoraInici*, *pHoraFi*, *pCurs*, *pAnyAcademic* i *pAssignatura*.
- Cap dels paràmetres pot ser nul.
- L'assignatura existeix a la taula *Assignatura*.
- El calendari escolar existeix a la taula *Calendari_Escolar*.

Postcondicions:

- Modifica l'assignatura del calendari escolar.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Calendari escolar modificat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.14. Alta alumne

Aquest procediment insereix un nou alumne a la taula *Alumne*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Alumne*.

Precondicions:

- Els paràmetres d'entrada són *pDNI*, *pExpedient*, *pNom*, *pCognom1*, *pCognom2*, *pDataNaixement*, *pAdreça*, *pCiutat*, *pCodiPostal*, *pProvíncia*, *pTelefon*, *pMobil* i *pEmail*.
- Els únics paràmetres que poden ser nuls són *pTelefon*, *pMobil* i *pEmail*.
- L'alumne no existeix a la taula *Alumne*.

Postcondicions:

- Afegeix un nou alumne a la taula *Alumne*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Alumne inserit.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.15. Baixa alumne

Aquest procediment elimina un alumne de la taula *Alumne*: agafa els paràmetres d'entrada i esborra totes les dades d'un alumne determinat amb una sentència DELETE a la taula *Alumne*.

Precondicions:

- El paràmetre d'entrada és *pAlumne*.
- El paràmetre no pot ser nul.
- L'alumne existeix a la taula *Alumne*.

Postcondicions:

- Esborra l'alumne de la taula *Alumne*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Alumne esborrat.'
 - Si l'execució finalitza amb un error: 'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.16. Modificació alumne

Aquest procediment modifica un alumne de la taula *Alumne*: agafa els paràmetres d'entrada i modifica totes les dades d'un alumne determinat amb una sentència UPDATE a la taula *Alumne*.

Precondicions:

- Els paràmetres d'entrada són *pDNI*, *pExpedient*, *pNom*, *pCognom1*, *pCognom2*, *pDataNaixement*, *pAdreça*, *pCiutat*, *pCodiPostal*, *pProvincia*, *pTelefon*, *pMobil* i *pEmail*.
- Els únics paràmetres que poden ser nuls són *pTelefon*, *pMobil* i *pEmail*.
- L'alumne existeix a la taula *Alumne*.

Postcondicions:

- Modifica l'expedient, el nom, el primer i segon cognom, la data de naixement, l'adreça, la ciutat, el codi postal, la província, el telèfon, el mòbil i l'email de l'alumne.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Alumne modificat.'
 - Si l'execució finalitza amb un error: 'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.17. Alta matrícula

Aquest procediment insereix una nova matrícula a la taula *Matricula*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Matricula*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pCurs* i *pAnyAcademic*.
- Cap dels paràmetres pot ser nul.
- L'alumne existeix a la taula *Alumne*.
- El curs i any acadèmic existeixen a la taula *Curs*.
- La matrícula no existeix a la taula *Matricula*.

Postcondicions:

- Afegeix una nova matrícula a la taula *Matricula*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Matrícula inserida.'

- Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.18. Baixa matrícula

Aquest procediment elimina una matrícula d'un alumne de la taula *Matricula*: agafa els paràmetres d'entrada i esborra totes les dades d'una matrícula determinada amb una sentència DELETE a la taula *Matricula*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pCurs* i *pAnyAcademic*.
- Cap dels paràmetres pot ser nul.
- La matrícula existeix a la taula *Matricula*.

Postcondicions:

- Esborra la matrícula de la taula *Matricula*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Matrícula esborrada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.19. Alta professor

Aquest procediment insereix un nou professor a la taula *Professor*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Professor*.

Precondicions:

- Els paràmetres d'entrada són *pDNI*, *pNom*, *pCognom1*, *pCognom2*, *pDataNaixement*, *pAdreça*, *pCiutat*, *pCodiPostal*, *pProvincia*, *pTelefon*, *pMobil* i *pEmail*.
- Els únics paràmetres que poden ser nuls són *pTelefon*, *pMobil* i *pEmail*.
- El professor no existeix a la taula *Professor*.

Postcondicions:

- Afegeix un nou professor a la taula *Professor*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Professor inserit.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.20. Baixa professor

Aquest procediment elimina un professor de la taula *Professor*: agafa els paràmetres d'entrada i esborra totes les dades d'un professor determinat amb una sentència DELETE a la taula *Professor*.

Precondicions:

- El paràmetre d'entrada és *pProfessor*.

- El paràmetre no pot ser nul.
- El professor existeix a la taula *Professor*.

Postcondicions:

- Esborra el professor de la taula *Professor*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Professor esborrat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.21. Modificació professor

Aquest procediment modifica un professor de la taula *Professor*: agafa els paràmetres d'entrada i modifica totes les dades d'un professor determinat amb una sentència UPDATE a la taula *Professor*.

Precondicions:

- Els paràmetres d'entrada són *pDNI*, *pNom*, *pCognom1*, *pCognom2*, *pDataNaixement*, *pAdreça*, *pCiutat*, *pCodiPostal*, *pProvíncia*, *pTelefon*, *pMobil* i *pEmail*.
- Els únics paràmetres que poden ser nuls són *pTelefon*, *pMobil* i *pEmail*.
- El professor existeix a la taula *Professor*.

Postcondicions:

- Modifica el nom, el primer i segon cognom, la data de naixement, l'adreça, la ciutat, el codi postal, la província, el telèfon, el mòbil i l'email del professor.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Professor modificat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.22. Alta impartir

Aquest procediment insereix una nova associació professor-assignatura a la taula *Impartir*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Impartir*.

Precondicions:

- Els paràmetres d'entrada són *pProfessor*, *pAssignatura*, *pcurs* i *pAnyAcademic*.
- Cap dels paràmetres pot ser nul.
- El professor existeix a la taula *Professor*.
- L'assignatura existeix a la taula *Assignatura*.
- El curs i any acadèmic existeixen a la taula *Curs*.
- L'associació professor-assignatura no existeix a la taula *Impartir*.

Postcondicions:

- Afegeix una nova associació professor-assignatura a la taula *Impartir*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Associació professor-assignatura inserida.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.

- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.23. Baixa impartir

Aquest procediment elimina una associació professor-assignatura de la taula *Impartir*: agafa els paràmetres d'entrada i esborra totes les dades d'una associació determinada amb una sentència DELETE a la taula *Impartir*.

Precondicions:

- Els paràmetres d'entrada són *pProfessor*, *pAssignatura*, *pcurs* i *pAnyAcademic*.
- Cap dels paràmetres pot ser nul.
- La associació professor-assignatura existeix a la taula *Impartir*.

Postcondicions:

- Esborra la associació professor-assignatura de la taula *Impartir*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Associació professor-assignatura esborrada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.24. Alta calendari tutoria

Aquest procediment insereix un nou calendari de tutoria als alumnes i als pares a la taula *Calendari_Tutoria*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Calendari_Tutoria*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pHoraInici*, *pHoraFi*, *pcurs*, *pAnyAcademic*, *pTutor* i *pHorariAlumnes*.
- L'únic paràmetre que pot ser nul és *pTutor*.
- El paràmetre *pHoraFi* ha de ser posterior al paràmetre *pHoraInici*.
- El professor existeix a la taula *Professor*.
- La data existeix a la taula *Calendari*.
- El curs i any acadèmic existeixen a la taula *Curs*.
- El calendari de tutoria no existeix a la taula *Calendari_Tutoria*.

Postcondicions:

- Afegeix un nou calendari a la taula *Calendari_Tutoria*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Calendari tutoria inserit.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.25. Baixa calendari tutoria

Aquest procediment elimina un calendari de tutoria als alumnes i als pares de la taula *Calendari_Tutoria*: agafa els paràmetres d'entrada i esborra totes les dades d'un calendari determinat amb una sentència DELETE a la taula *Calendari_Tutoria*

Precondicions:

- Els paràmetres d'entrada són *pData*, *pHoraInici*, *pHoraFi*, *pCurs* i *pAnyAcademic*.
- L'únic paràmetre que pot ser nul és *pTutor*.
- El calendari de tutoria existeix a la taula *Calendari_Tutoria*.

Postcondicions:

- Esborra el calendari de la taula *Calendari_Tutoria*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Calendari tutoria esborrat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.26. Modificació calendari tutoria

Aquest procediment modifica un calendari de tutoria als alumnes i als pares de la taula *Calendari_Tutoria*: agafa els paràmetres d'entrada i els modifica totes les dades d'un calendari determinat amb una sentència UPDATE a la taula *Calendari_Tutoria*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pHoraInici*, *pHoraFi*, *pcurs*, *pAnyAcademic*, *pTutor* i *pHorariAlumnes*.
- L'únic paràmetre que pot ser nul és *pTutor*.
- El calendari de tutoria existeix a la taula *Calendari_Tutoria*.

Postcondicions:

- Modifica les dades del calendari de tutoria.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Calendari tutoria modificat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.27. Alta reunió alumne

Aquest procediment insereix una nova reunió amb els alumnes a la taula *Reunio_Alumne*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Reunio_Alumne*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pProfessor*, *pDataiHora* i *pTema*.
- L'únic paràmetre que pot ser nul és *pTema*.
- L'alumne existeix a la taula *Alumne*.
- El professor existeix a la taula *Professor*.
- La reunió amb l'alumne no existeix a la taula *Reunio_Alumne*.

Postcondicions:

- Afegeix una nova reunió a la taula *Reunio_Alumne*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Reunió amb l'alumne inserida.'

- Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.28. Baixa reunió alumnes

Aquest procediment elimina una reunió amb els alumnes de la taula *Reunio_Alumne*: agafa els paràmetres d'entrada i esborra totes les dades d'un horari de consulta determinat amb una sentència DELETE a la taula *Reunio_Alumne*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pProfessor*, *pDataiHora*.
- Cap dels paràmetres pot ser nul.
- La reunió existeix a la taula *Reunio_Alumne*.

Postcondicions:

- Esborra la reunió de la taula *Reunio_Alumne*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Reunió amb l'alumne esborrada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.29. Modificació reunió alumne

Aquest procediment modifica una reunió amb un alumne de la taula *Reunio_Alumne*: agafa els paràmetres d'entrada i modifica totes les dades d'una reunió determinada amb una sentència UPDATE a la taula *Reunio_Alumne*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pProfessor*, *pDataiHora* i *pTema*.
- L'únic paràmetre que pot ser nul és *pTema*.
- La reunió amb l'alumne existeix a la taula *Reunio_Alumne*.

Postcondicions:

- Modifica el tema de la reunió.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Reunió amb l'alumne modificada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.30. Alta reunió pares

Aquest procediment insereix una nova reunió amb els pares dels alumnes a la taula *Reunio_Pares*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Reunio_Pares*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pProfessor*, *pDataiHora* i *pTema*.
- L'únic paràmetre que pot ser nul és *pTema*.
- L'alumne existeix a la taula *Alumne*.

- El professor existeix a la taula *Professor*.
- La reunió amb els pares de l'alumne no existeix a la taula *Reunio_Pares*.

Postcondicions:

- Afegeix una nova reunió a la taula *Reunio_Pares*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Reunió amb els pares de l'alumne inserida.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.31. Baixa reunió pares

Aquest procediment elimina una reunió amb els pares dels alumnes de la taula *Reunio_Pares*: agafa els paràmetres d'entrada i esborra totes les dades d'una reunió amb els pares determinada amb una sentència DELETE a la taula *Reunio_Pares*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pProfessor*, *pData* i *pHora*.
- Cap dels paràmetres pot ser nul.
- La reunió existeix a la taula *Reunio_Pares*.

Postcondicions:

- Esborra la reunió de la taula *Reunio_Pares*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Reunió amb els pares de l'alumne esborrada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.32. Modificació reunió pares

Aquest procediment modifica una reunió amb els pares d'un alumne de la taula *Reunio_Pares*: agafa els paràmetres d'entrada i modifica totes les dades d'una reunió determinada amb una sentència UPDATE a la taula *Reunio_Pares*.

Precondicions:

- Els paràmetres d'entrada són *pAlumne*, *pProfessor*, *pData* i *Hora* i *pTema*.
- L'únic paràmetre que pot ser nul és *pTema*.
- La reunió amb els pares de l'alumne existeix a la taula *Reunio_Pares*.

Postcondicions:

- Modifica el tema de la reunió.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Reunió amb els pares de l'alumne modificada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.33. Alta tipus amonestació

Aquest procediment insereix un nou tipus d'amonestació a la taula *Tipus_Amonestacio*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Tipus_Amonestacio*.

Precondicions:

- Els paràmetres d'entrada són *pTipus* i *pDescripcio*.
- Cap dels paràmetres pot ser nul.
- El tipus d'amonestació no existeix a la taula *Tipus_Amonestacio*.

Postcondicions:

- Afegeix un nou tipus d'amonestació a la taula *Tipus_Amonestacio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Tipus d'amonestació inserit.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.34. Baixa tipus amonestació

Aquest procediment elimina un tipus d'amonestació de la taula *Tipus_Amonestacio*: agafa els paràmetres d'entrada i esborra totes les dades d'un tipus d'amonestació determinat amb una sentència DELETE a la taula *Tipus_Amonestacio*.

Precondicions:

- El paràmetre d'entrada és *pTipus*.
- El paràmetre no pot ser nul.
- El tipus d'amonestació existeix a la taula *Tipus_Amonestacio*.

Postcondicions:

- Esborra el tipus d'amonestació de la taula *Tipus_Amonestacio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Tipus d'amonestació esborrat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.35. Modificació tipus amonestació

Aquest procediment modifica un tipus d'amonestació de la taula *Tipus_Amonestacio*: agafa els paràmetres d'entrada i modifica totes les dades d'un tipus d'amonestació determinat amb una sentència UPDATE a la taula *Tipus_Amonestacio*.

Precondicions:

- Els paràmetres d'entrada són *pIdTipus*, *pTipus* i *pDescripcio*.
- Cap dels paràmetres pot ser nul.
- El tipus d'amonestació existeix a la taula *Tipus_Amonestacio*.

Postcondicions:

- Modifica el tipus, la descripció i la sanció del tipus d'amonestació.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Tipus d'amonestació modificat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.

- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.36. Alta tipus sanció

Aquest procediment insereix un nou tipus de sanció a la taula *Tipus_Sancio*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Tipus_Sancio*.

Precondicions:

- Els paràmetres d'entrada són *pTipus*, *pActivacio*, *pTipusAmonestacio* i *pValor*.
- Cap dels paràmetres pot ser nul.
- El tipus d'amonestació existeix a la taula *Tipus_Amonestacio*.
- El tipus de sanció no existeix a la taula *Tipus_Sancio*.

Postcondicions:

- Afegeix un nou tipus de sanció a la taula *Tipus_Sancio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Tipus de sanció inserit.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.37. Baixa tipus sanció

Aquest procediment elimina un tipus de sanció de la taula *Tipus_Sancio*: agafa els paràmetres d'entrada i esborra totes les dades d'un tipus de sanció determinat amb una sentència DELETE a la taula *Tipus_Sancio*.

Precondicions:

- El paràmetre d'entrada és *pTipus*.
- El paràmetre no pot ser nul.
- El tipus de sanció existeix a la taula *Tipus_Sancio*.

Postcondicions:

- Esborra el tipus de sanció de la taula *Tipus_Sancio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Tipus de sanció esborrat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.38. Modificació tipus sanció

Aquest procediment modifica un tipus de sanció de la taula *Tipus_Sancio*: agafa els paràmetres d'entrada i modifica totes les dades d'un tipus de sanció determinat amb una sentència UPDATE a la taula *Tipus_Sancio*.

Precondicions:

- Els paràmetres d'entrada són *pTipus*, *pActivacio*, *pTipusAmonestacio* i *pValor*.
- El tipus d'amonestació existeix a la taula *Tipus_Amonestacio*.
- Cap dels paràmetres pot ser nul.

- El tipus de sanció existeix a la taula *Tipus_Sancio*.

Postcondicions:

- Modifica el tipus de sanció de la taula *Tipus_Sancio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Tipus de sanció modificat.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.39. Alta amonestació

Aquest procediment insereix una nova amonestació a la taula *Amonestacio*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Amonestacio*.

Precondicions:

- Els paràmetres d'entrada són *pDataiHora*, *pComunicacioPares*, *pGravetat*, *pAmonestat*, *pAmonestador*, *pCurs*, *pAnyAcademic*, *pAssignatura* i *pTipusAmonestacio*.
- Cap dels paràmetres pot ser nul.
- El paràmetre *pComunicacioPares* només pot prendre els valors "s" o "n".
- L'alumne existeix a la taula *Alumne*.
- El professor existeix a la taula *Professor*.
- El curs i any acadèmic existeixen a la taula *Curs*.
- L'assignatura existeix a la taula *Assignatura*.
- El tipus d'amonestació existeix a la taula *Tipus_Amonestacio*.
- L'amonestació no existeix a la taula *Amonestacio*.

Postcondicions:

- Afegeix un nova amonestació a la taula *Amonestacio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Amonestació inserida.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.40. Baixa amonestació

Aquest procediment elimina una amonestació de la taula *Amonestacio*: agafa els paràmetres d'entrada i esborra totes les dades d'una amonestació determinada amb una sentència DELETE a la taula *Amonestacio*.

Precondicions:

- El paràmetre d'entrada és *pAmonestacio*.
- El paràmetre no pot ser nul.
- L'amonestació existeix a la taula *Amonestacio*.

Postcondicions:

- Esborra l'amonestació de la taula *Amonestacio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Amonestació esborrada.'

- Si l'execució finalitza amb un error:
 'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.41. Modificació amonestació

Aquest procediment modifica una amonestació de la taula *Amonestacio*: agafa els paràmetres d'entrada i modifica totes les dades d'una amonestació determinada amb una sentència UPDATE a la taula *Amonestacio*.

Precondicions:

- Els paràmetres d'entrada són *pIdAmonestacio*, *pDataiHora*, *pComunicacioPares*, *pGravetat*, *pAmonestat*, *pAmonestador*, *pCurs*, *pAnyAcademic*, *pAssignatura* i *pTipusAmonestacio*.
- Cap dels paràmetres pot ser nul.
- El paràmetre *pComunicacioPares* només pot prendre els valors "s" o "n".
- L'alumne existeix a la taula *Alumne*.
- El professor existeix a la taula *Professor*.
- El curs i any acadèmic existeixen a la taula *Curs*.
- L'assignatura existeix a la taula *Assignatura*.
- El tipus d'amonestació existeix a la taula *Tipus_Amonestacio*.
- L'amonestació existeix a la taula *Amonestacio*.

Postcondicions:

- Modifica la data, l'hora, la comunicació amb els pares, la gravetat, l'alumne amonestat, l'alumne amonestador, el curs, l'any acadèmic, l'assignatura i el tipus d'amonestació de l'amonestació.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Amonestació modificada.'
 - Si l'execució finalitza amb un error:
 'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.41. Alta sanció

Aquest procediment insereix una nova sanció a la taula *Sancio*: agafa els paràmetres d'entrada i els insereix directament amb una sentència INSERT a la taula *Sancio*.

Precondicions:

- Els paràmetres d'entrada són *pData*, *pMotiu*, *pResolucio*, *pSancionat*, *pCurs*, *pAnyAcademic* i *pTipusSancio*.
- Cap dels paràmetres pot ser nul.
- L'alumne existeix a la taula *Alumne*.
- El tipus de sanció existeix a la taula *Tipus_Sancio*.
- La sanció no existeix a la taula *Sancio*.

Postcondicions:

- Afegeix una nova sanció a la taula *Sancio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Sanció inserida.'
 - Si l'execució finalitza amb un error:

'ERROR' + codi error SQL + missatge error SQL.

- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.42. Baixa sanció

Aquest procediment elimina una sanció de la taula *Sancio*: agafa els paràmetres d'entrada i esborra totes les dades d'una sanció determinat amb una sentència DELETE a la taula *Sancio*.

Precondicions:

- El paràmetre d'entrada és *pSancio*.
- El paràmetre no pot ser nul.
- La sanció existeix a la taula *Sancio*.

Postcondicions:

- Esborra la sanció de la taula *Sancio*.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Sanció esborrada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.43. Modificació sanció

Aquest procediment modifica una sanció de la taula *Sancio*: agafa els paràmetres d'entrada i modifica totes les dades d'una sanció determinada amb una sentència UPDATE a la taula *Sancio*.

Precondicions:

- Els paràmetres d'entrada són *pIdSancio*, *pData*, *pMotiu*, *pResolucio*, *pSancionat* i *pTipusSancio*.
- Cap dels paràmetres pot ser nul.
- L'alumne existeix a la taula *Alumne*.
- El tipus de sanció existeix a la taula *Tipus_Sancio*.
- La sanció existeix a la taula *Sancio*.

Postcondicions:

- Modifica la data, el motiu, la resolució, l'alumne sancionat i el tipus de sanció de la sanció.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Sanció modificada.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.44. Consulta curs de l'alumne

Aquest procediment permet saber a quin curs pertany un alumne en un any acadèmic determinat: agafa els paràmetres d'entrada i busca a la taula *MATRICULA* l'alumne amb el mateix DNI i any acadèmic entrats com a paràmetres.

Precondicions:

- Els paràmetres d'entrada són *pAlumne* i *pAnyAcademic*.

- Els paràmetres no poden ser nuls.
- L'alumne està matriculat en l'any acadèmic entrat com a paràmetre.

Postcondicions:

- Mostra l'identificador i el nom del curs.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta curs alumne.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.46. Consulta professors de les assignatures

Aquest procediment permet saber quins professors donen una assignatura en un any acadèmic determinat: agafa els paràmetres d'entrada i busca a la taula *IMPARTIR* el professor amb el mateix DNI i any acadèmic entrats com a paràmetres.

Precondicions:

- Els paràmetres d'entrada són *pAssignatura* i *pAnyAcademic*.
- Els paràmetres no poden ser nuls.
- L'assignatura s'imparteix en l'any acadèmic entrat com a paràmetre.

Postcondicions:

- Mostra el DNI, el nom i cognoms dels professor.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta professors assignatura.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.47. Consulta tutor del curs

Aquest procediment permet saber quin professor és el tutor d'un curs en un any acadèmic determinat: agafa els paràmetres d'entrada i busca a la taula *CURS* el professor amb el mateix DNI i any acadèmic entrats com a paràmetres.

Precondicions:

- Els paràmetres d'entrada són *pCurs* i *pAnyAcademic*.
- Els paràmetres no poden ser nuls.
- El curs s'imparteix en l'any acadèmic entrat com a paràmetre.

Postcondicions:

- Mostra el DNI, el nom i cognoms del tutor.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta tutor curs.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.48. Consulta amonestacions i sancions de l'alumne

Aquest procediment permet saber les amonestacions i sancions imposades a un alumne en un any acadèmic determinat: agafa els paràmetres d'entrada i busca a les taules AMONESTACIO i SANCIO l'alumne amb el mateix DNI i any acadèmic entrats com a paràmetres.

Precondicions:

- Els paràmetres d'entrada són *pAlumne* i *pAnyAcademic*.
- Els paràmetres no poden ser nuls.
- L'alumne està matriculat en l'any acadèmic entrat com a paràmetre.

Postcondicions:

- Mostra els identificadors, els tipus i les data de les amonestacions i els identificadors, els tipus i les dates de les sancions imposades a l'alumne.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta amonestacions i sancions dels alumnes.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.49. Llistat amonestacions imposades

Aquest procediment permet saber les amonestacions imposades als alumnes.

Precondicions:

- No rep cap paràmetre d'entrada.

Postcondicions:

- Mostra els identificadors, els tipus, data, curs, any acadèmic, DNI, nom i cognoms de l'alumne.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Llistat amonestacions.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.50. Llistat alumnes del curs

Aquest procediment permet saber tots els alumnes d'un curs en un any acadèmic determinat: agafa els paràmetres d'entrada i busca a la taula MATRICULA el curs amb el mateix identificador i any acadèmic entrats com a paràmetres.

Precondicions:

- Els paràmetres d'entrada són *pCurs* i *pAnyAcademic*.
- Els paràmetres no poden ser nuls.
- El curs s'imparteix en l'any acadèmic entrat com a paràmetre.

Postcondicions:

- Mostra el DNI, el nom i cognoms dels alumnes.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Llistat alumnes curs.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.

- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.51. Llistat amonestacions i sancions

Aquest procediment permet saber les amonestacions i sancions disponibles per a aplicar en el centre.

Precondicions:

- No hi ha cap paràmetre d'entrada.

Postcondicions:

- Mostra els identificadors, els tipus i les descripcions de les amonestacions i els identificadors, els tipus i el moment d'activació de les sancions.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Llistat amonestacions i sancions.'
 - Si l'execució finalitza amb un error:
'ERROR' + nom del procediment + descripció de l'error.
'ERROR' + traça + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.52. Llistat amonestacions i sancions de l'alumne

Aquest procediment permet saber les amonestacions i sancions imposades a un: agafa els paràmetres d'entrada i busca a les taules AMONESTACIO i SANCIO l'alumne amb el mateix DNI entrat com a paràmetre.

Precondicions:

- El paràmetre d'entrada és *pAlumne*.
- El paràmetre *pAlumne* no pot ser nul.
- L'alumne existeix en la taula *Matricula*.

Postcondicions:

- Mostra els identificadors, els tipus i les dates de les amonestacions i els identificadors, els tipus i les dates de les sancions imposades a l'alumne.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Llistat amonestacions i sancions dels alumnes.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.53. Consulta número d'amonestacions alumne

Aquest procediment forma part del mòdul estadístic i permet saber el número d'amonestacions per alumne.

Precondicions:

- No hi ha cap paràmetre d'entrada.

Postcondicions:

- Mostra el número d'amonestacions i el DNI, nom i cognoms de l'alumne.
- El paràmetre de sortida *RSP* tindrà el valor:

- Si l'execució finalitza amb èxit: 'OK: Consulta número amonestacions per alumne.'
- Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.54. Consulta número de sancions alumne i any

Aquest procediment forma part del mòdul estadístic i permet saber el número de sancions per alumne i any acadèmic.

Precondicions:

- No hi ha cap paràmetre d'entrada.

Postcondicions:

- Mostra el número de sancions i el DNI, nom i cognoms de l'alumne i l'any acadèmic.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta número sancions per alumne i any.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.55. Consulta mitjana d'amonestacions professor i any

Aquest procediment forma part del mòdul estadístic i permet saber la mitjana d'amonestacions per professor i any.

Precondicions:

- No hi ha cap paràmetre d'entrada.

Postcondicions:

- Mostra la mitjana d'amonestacions i el DNI, nom i cognoms del professor i l'any acadèmic.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta mitjana amonestacions per professor i any.'
 - Si l'execució finalitza amb un error:
'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.56. Consulta número de sancions curs i any

Aquest procediment forma part del mòdul estadístic i permet saber el número de sancions per curs i any acadèmic.

Precondicions:

- No hi ha cap paràmetre d'entrada.

Postcondicions:

- Mostra el número de sancions i l'identificador i el nom del curs i l'any acadèmic.
- El paràmetre de sortida *RSP* tindrà el valor:

- Si l'execució finalitza amb èxit: 'OK: Consulta número sancions per curs i any.'
- Si l'execució finalitza amb un error:
 - 'ERROR' + nom del procediment + descripció de l'error.
 - 'ERROR' + traça + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.57. Consulta alumne més sancionat any

Aquest procediment forma part del mòdul estadístic i permet saber l'alumne més sancionat en un any determinat.

Precondicions:

- El paràmetre d'entrada és *pAnyAcademic*.
- El paràmetre *pAlumne* no pot ser nul.

Postcondicions:

- Mostra el número de sancions i el DNI, nom i cognoms de l'alumne.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta alumne més sancionat en l'any.'
 - Si l'execució finalitza amb un error:
 - 'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.58. Consulta professor més amonestador curs

Aquest procediment forma part del mòdul estadístic i permet saber el professor més amonestador per curs.

Precondicions:

- No hi ha cap paràmetre d'entrada.

Postcondicions:

- Mostra el número de sancions i el DNI, nom i cognoms del professor i l'identificador i el nom del curs.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta professor més amonestador per curs.'
 - Si l'execució finalitza amb un error:
 - 'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.2.59. Consulta mitjana de sancions curs

Aquest procediment forma part del mòdul estadístic i permet saber la mitjana de sancions per curs.

Precondicions:

- No hi ha cap paràmetre d'entrada.

Postcondicions:

- Mostra la mitjana de sancions i l'identificador i el nom del curs.
- El paràmetre de sortida *RSP* tindrà el valor:
 - Si l'execució finalitza amb èxit: 'OK: Consulta mitjana sancions per curs.'

- Si l'execució finalitza amb un error:
 'ERROR' + codi error SQL + missatge error SQL.
- Insereix a la taula *Log_Taula* el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

5. CONCLUSIONS

Com a conclusió general, es pot dir que s'han assolit els objectius marcats a l'inici del projecte, tant els objectius generals com els específics.

La part més complicada del projecte ha estat l'utilització del gestor de bases de dades Oracle i el llenguatge de programació PL/SQL, degut a la poca experiència prèvia amb aquest programari. Per tant, tots dos han representat un esforç important d'estudi.

La part més complicada ha estat el disseny i la implementació dels procediments, ja que les dades havien de complir uns requisits determinats i s'havia de conèixer bé el llenguatge de programació.

Finalment, s'hauria d'haver fet un joc de proves per a comprovar el funcionament del sistema, però per falta de temps no m'ha donat temps a fer-lo. En tot cas, si que s'ha comprovat el funcionament del sistema individualment, a mida que s'anaven fent cada taula i procediment.

6. GLOSSARI

Alumne

Persona física que s'ha donat d'alta en el sistema com alumne i que pot matricular-se en els cursos.

Amonestació

Correcció de caràcter lleu que s'imposa per infraccions de disciplina.

Assignatura

Conjunt de coneixements relacionats amb un tema i que s'imparteixen al llarg dels diferents cursos escolars.

Calendari escolar

Període que dura un curs al llarg de l'any, sol abarcar de setembre a juliol. Aquest calendari inclou l'horari de cada assignatura que forma cada curs.

Calendari tutoria

Període que dura un curs al llarg de l'any, sol abarcar de setembre a juliol. Aquest calendari inclou l'horari que té cada tutor per a atendre els alumnes i els pares dels alumnes.

Curs

Un curs està format per un grup d'assignatures.

Matricula

Assignació d'un alumne a un curs, en un període determinat.

Professor

Persona física que imparteix una o més assignatures.

Sanció

Càstig que prové de l'incompliment de les normes.

Sistema de Gestió de Bases de Dades (SGBD)

Conjunt de programari dissenyat per a facilitar la gestió d'un conjunt de dades en una base de dades.

Structured Query Language (SQL)

Llenguatge estàndard de consulta a bases de dades relacionals.

7. BIBLIOGRAFIA

Oracle 10g versió 10.2.0.1. Documentación Cd-Rom.

Scott Urman. "*Oracle 9i: Programación PL/SQL*". Oracle Press Osborne, Mc Graw Hill, 2002.

Sistac Planas, Jaume (2005). Bases de dades I. Barcelona: Eureka Media SL.

Sistac Planas, Jaume (2004). Bases de dades II. Barcelona: Eureka Media SL.