Disseny i implementació d'un sistema de gestió d'amonestacions i sancions en centres educatius

Memòria

Estudiant: Antònia Algaba Caballero **(ETIG)**

Consultor: Manel Rella Ruíz

12 de juny de 2011

ÍNDEX

1. INTRODUCCIÓ

Dins de l'objectiu general de l'assignatura, enfocat a desenvolupar un treball final de carrera dels estudis d'Enginyeria Tècnica d'Informàtica de Gestió (ETIG), aquest projecte pretén consolidar, aprofundir i posar en pràctica els coneixements adquirits, especialment pel que fa a les assignatures relatives a l'àrea de bases de dades. Així, simulant l'entorn d'un projecte real, es pot comprovar la capacitat de gestionar i executar un projecte en totes les seves fases, d'acord amb un pla de treball i seguint una planificació.

El projecte consisteix a dissenyar i implementar una base de dades relacional que, mitjançant procediments emmagatzemats, permeti definir les funcionalitats principals del sistema: inserir, esborrar i/o modificar dades; a més, ha de proporcionar diversa informació estadística.

El present document té per objecte presentar el treball que s'ha realitzat per a assolir el projecte, recollint de forma concreta les tasques dutes a terme durant les diferents etapes de què ha constat: el pla de treball, l'anàlisi dels requeriments, el disseny, la implementació de la base de dades i el *testing* final del projecte.

(índex)

2. DESCRIPCIÓ GENERAL DEL PROJECTE

2.1. Objectius

De forma general, el treball s'ha centrat en el desenvolupament d'un sistema informàtic que pogués servir com a magatzem d'informació per a una futura aplicació de gestió d'amonestacions i sancions en centres educatius d'ensenyament secundari dependents de la Generalitat de Catalunya i ha implicat tant el disseny com la implementació de la base de dades; l'aplicació concreta de gestió, però, ha restat fora de l'àmbit d'aquest treball.

El sistema, que es gestiona a través de cadascun dels diferents instituts d'ensenyament de Catalunya, recull informació bàsica dels alumnes matriculats, dels cursos en què s'han matriculat i dels professors que els imparteixen i permet introduir-hi les incidències i/o amonestacions relacionades amb cada un d'ells, així com també les sancions que se'n derivin, si s'escau.

La base de dades dissenyada permet, mitjançant els procediments emmagatzemats oportuns, inserir, modificar i suprimir dades relatives als alumnes, als professors, als cursos, a les assignatures, al calendari escolar, a les amonestacions i a les sancions,

així com efectuar diferents tipus de consultes; a més, porta associat un mòdul estadístic que proporciona informació actualitzada en tot moment.

(índex)

2.2. Metodologia

Per a assolir els objectius marcats, la metodologia que s'ha seguit correspon a l'anomenat *desenvolupament en cascada*, de manera que, inicialment, s'ha dut a terme una fase d'anàlisi amb la recollida de requisits, seguida d'una fase de disseny de la base de dades requerida, per continuar després amb la implementació i, finalment, amb la realització de les proves de *testing*.

Així, doncs, el treball ha requerit realitzar els diagrames E/R corresponents, construir els scripts de creació de taules i seqüències i implementar els procediments necessaris. Aquests procediments, a més d'especificar-se amb la informació suficient per tal que puguin ser utilitzats pels programadors de la capa de presentació, compleixen les condicions següents:

- Defineixen un paràmetre de sortida que indica com ha finalitzat l'execució.
- Disposen del tractament d'excepcions.
- Emmagatzemen totes les crides a procediments en una taula de log.

Per últim, s'ha implementat la inicialització de la base de dades amb un conjunt de dades suficients i es presenta un joc de proves que garanteix el bon funcionament de les funcionalitats implementades, així com el control d'errors i d'excepcions.

(índex)

2.3. Recursos utilitzats

2.3.1. Humans

Atès el caràcter individual de l'assignatura, totes les tasques dels participants habituals d'un projecte, pel que fa al proveïdor, convergeixen en una única persona, l'estudiant, que ha assolit tant les tasques de directora del projecte com les dels desenvolupadors del producte. Així, ha estat una única persona l'encarregada de realitzar el disseny conceptual, d'instal·lar el programari necessari i d'implementar i executar directament les funcionalitats del projecte.

El consultor de l'assignatura, per la seva banda, ha exercit les tasques pròpies del client.

2.3.2. Maquinari

El maquinari de què s'ha disposat per dur a terme el projecte ha estat el següent:

- Ordinador portàtil ACER ASPIRE 5738ZG, amb processador Intel Pentium T4400 (2.2 GHz), 4 Gb de memòria DDR, 640 GB de disc dur i sistema operatiu Windows 7 Home Premium

2.3.3. Programari

Pel que fa al programari, i d'acord amb algunes de les especificacions expresses del client, el projecte s'ha desenvolupat fent servir el que es detalla tot seguit:

- Sistema de gestió de base de dades: Oracle Express v10.2.0.1
- Desenvolupament SQL: SQL Developer v1.0.0.15
- Diagrama de Gantt: lGantt-0.4
- Documents de text: Microsoft Word 2010 (convertits després a pdf)
- Presentació: Microsoft PowerPoint 2010

(índex)

2.4. Temporització

El marc temporal en què s'ha dut a terme el projecte ha estat marcat per uns terminis prefixats pel propi client i ha estat comprès entre el 3 de març de 2011, data de presentació de l'enunciat del projecte, i el 12 de juny de 2011, data final de lliurament. Els lliuraments parcials que s'han efectuat han tingut lloc el 20 de març de 2011 (lliurament del Pla de treball), el 17 d'abril de 2011 (entrega de l'anàlisi de requisits i del disseny del projecte) i 15 de maig de 2011 (lliurament de la implementació del projecte).

(índex)

2.5. Anàlisi de riscos

El principal risc esdevingut ha estat marcat per la simultaneïtat en la realització del present projecte amb el desenvolupat per a un altre client (assignatura: Criptografia). La coincidència en el temps dels lliuraments d'ambdós projectes, però, sobretot, el desconeixement de la dificultat i del contingut de la pràctica que he hagut de dur a terme, va fer que no pogués planificar molt acuradament la distribució del temps de què disposava, la qual cosa ha comportat que la planificació presentada en aquest pla de treball es modifiqués lleugerament quant al desenvolupament de les tasques intermèdies previstes, tot mantenint, però, les dates claus de lliurament de les diferents PAC i del projecte final.

La planificació prevista no s'ha vist afectada per incidents a nivell tècnic, per fallada del maquinari o del programari que he fet servir (un problema amb què m'havia trobat en alguna altra assignatura anterior). Tot i així, en previsió de què pogués passar, he anat

fent una còpia de tot el projecte en un altre suport diferent (extern) en finalitzar cada sessió de treball.

Per últim, sí m'he vist afectada per una situació personal força greu que va tenir especial incidència en el moment de fer el tercer lliurament relatiu al projecte. El control d'aquesta situació restava fora del meu abast.

(índex)

2.6. Estructura del projecte

D'acord amb els requeriments marcats, les diferents fases en què s'ha estructurat el projecte, així com les activitats incloses en cada una d'elles, s'especifiquen a continuació:

PAC 1: Estudi del projecte.
Instal·lació del programari.
Elaboració i lliurament del Pla de treball.

PAC 2: Estudi amb profunditat dels requeriments del sistema.
Disseny conceptual de la base de dades.
Disseny lògic de la base de dades.
Implementació: Inici d'elaboració dels *scripts* de les taules.
Inici del disseny de la Memòria final.
Lliurament PAC 2: diagrames E/R i scripts inicials de creació de taules.

PAC 3: Implementació: revisió i finalització dels *scripts* de creació de taules.
Elaboració de seqüències, disparadors, funcions i procediments emmagatzemats.
Seguiment de l'elaboració de la Memòria final del projecte.
Lliurament PAC 3: *scripts* finals de la base de dades (producte).

PAC 4: Elaboració de proves de *testing* i revisió final del producte.
Finalització de la Memòria del projecte.
Elaboració de la presentació del projecte.
Lliurament final del projecte: memòria, presentació i producte.

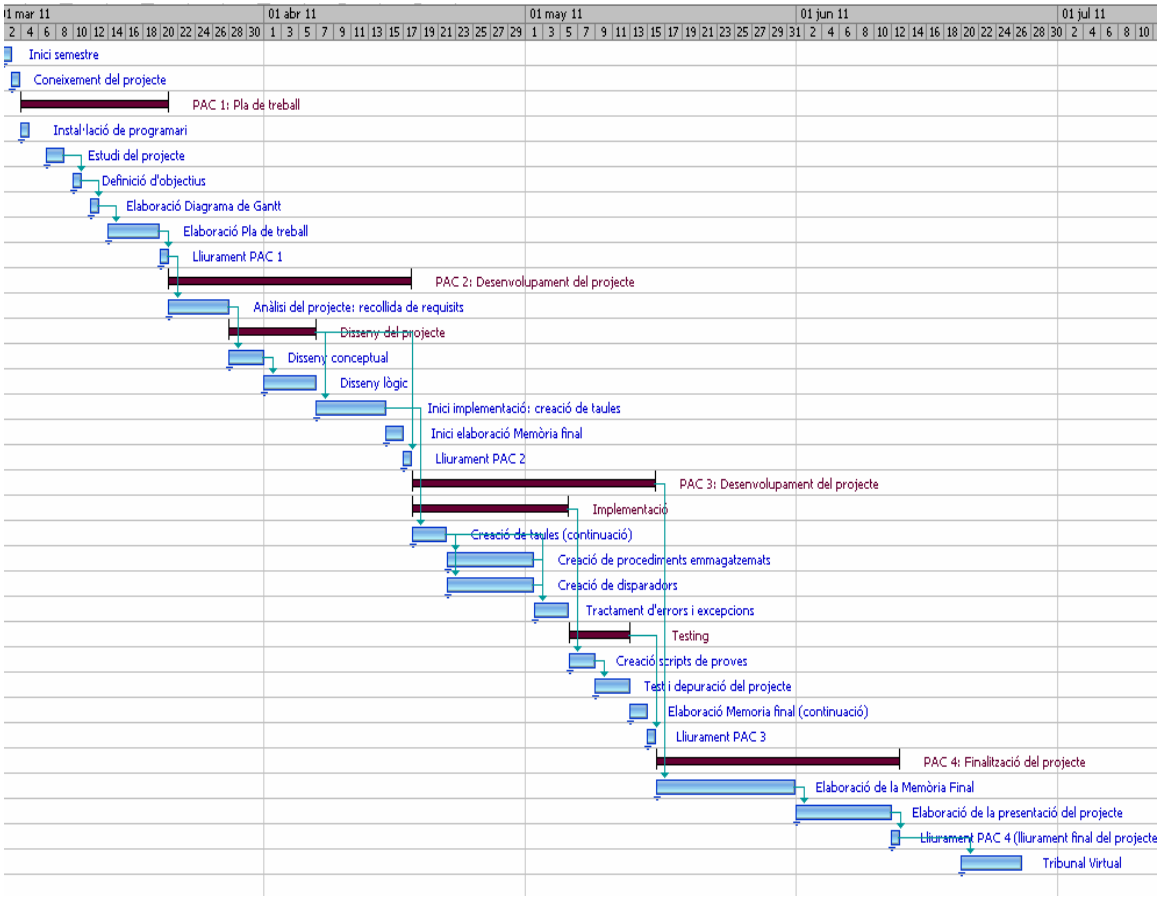
Tot i que, tal com s'ha assenyalat anteriorment, s'ha dut a terme un desenvolupament en cascada, els lliuraments parcials no han tingut correspondència estricta amb cada una de les fases d'aquesta metodologia. Així, a la PAC 2 es va fer un treball tant d'anàlisi detallat dels requeriments del projecte, com de disseny, principalment, fent-ne el disseny conceptual i transformant-lo al disseny lògic; també es va iniciar la implementació del projecte. A la PAC 3, per la seva banda, es va dur a terme la implementació, pròpiament, de la base de dades. Finalment, assenyalar que tot i que,

inicialment, s'havia previst crear i realitzar les proves de *testing* en el lliurament corresponent a la PAC 3, els problemes sobrevinguts explicats anteriorment van fer que es fessin amb posterioritat.

(índex)

2.7. Planificació del projecte: Diagrama de Gantt

El diagrama de Gantt següent mostra el detall de les tasques planificades en funció dels terminis marcats pel propi projecte:



Cal assenyalar que la planificació prevista s'ha seguit estrictament i s'han realitzat totes les tasques d'acord amb el calendari marcat, a excepció de la depuració del projecte amb les proves de *testing*, tal com ja s'ha explicat amb anterioritat.

(índex)

- *Especificació de les tasques*:

#	Name	Start	End	Length
1	Inici semestre	mié, 2/03	mié, 2/03	1d
2	Coneixement del projecte	jue, 3/03	jue, 3/03	1d
3	**PAC 1: Pla de treball**	**vie, 4/03**	**dom, 20/03**	**17d**
4	Instal·lació de programari	vie, 4/03	vie, 4/03	1d
5	Estudi del projecte	lun, 7/03	mar, 8/03	2d
6	Definició d'objectius	jue, 10/03	jue, 10/03	1d
7	Elaboració Diagrama de Gantt	sáb, 12/03	sáb, 12/03	1d
8	Elaboració del Pla de treball	lun, 14/03	sáb, 19/03	6d
9	Lliurament PAC 1	dom, 20/03	dom, 20/03	1d
10	**PAC 2: Desenvolupament del projecte**	**lun, 21/03**	**dom, 17/04**	**28d**
11	Anàlisi del projecte: recollida de requisits	lun, 21/03	dom, 27/03	7d
12	**Disseny del projecte**	**lun, 28/03**	**mié, 6/04**	**10d**
13	Disseny conceptual	lun, 28/03	jue, 31/03	4d
14	Disseny lògic	vie, 1/04	mié, 6/04	6d
15	Inici implementació: creació de taules	jue, 7/04	jue, 14/04	8d
16	Inici elaboració Memòria final	vie, 15/04	sáb, 16/04	2d
17	Lliurament PAC 2	dom, 17/04	dom, 17/04	1d
18	**PAC 3: Desenvolupament del projecte**	**lun, 18/04**	**dom, 15/05**	**28d**
19	**Implementació**	**lun, 18/04**	**jue, 5/05**	**18d**
20	Creació de taules (continuació)	lun, 18/04	jue, 21/04	4d
21	Creació de procediments emmagatzemats	vie, 22/04	dom, 1/05	10d
22	Creació de disparadors	vie, 22/04	dom, 1/05	10d
23	Tractament d'errors i excepcions	lun, 2/05	jue, 5/05	4d
24	**Testing**	**vie, 6/05**	**jue, 12/05**	**7d**
25	Creació scripts de proves	vie, 6/05	dom, 8/05	3d
26	Test i depuració del projecte	lun, 9/05	jue, 12/05	4d
27	Elaboració Memòria final (continuació)	vie, 13/05	sáb, 14/05	2d
28	Lliurament PAC 3	dom, 15/05	dom, 15/05	1d
29	**PAC 4: Finalització del projecte**	**lun, 16/05**	**dom, 12/06**	**28d**
30	Elaboració de la Memòria Final	lun, 16/05	mar, 31/05	16d
31	Elaboració de la presentació del projecte	mié, 1/06	sáb, 11/06	11d
32	Lliurament PAC 4 (lliurament del projecte final)	dom, 12/06	dom, 12/06	1d
33	Tribunal Virtual	lun, 20/06	dom, 26/06	7d

(índex)

3. ANÀLISI DE REQUISITS

La Conselleria d'Ensenyament de la Generalitat de Catalunya ha obert un concurs públic per rebre propostes de disseny d'una base de dades que els hi serveixi de magatzem d'informació per a una futura aplicació de gestió d'amonestacions i sancions en els centres educatius d'ensenyament secundari.

Aquesta base de dades ha d'emmagatzemar dades bàsiques relatives als alumnes, als cursos en què estan matriculats i als professors que els imparteixen, així com als diferents tipus d'amonestacions i sancions a què es poden veure sotmesos. De forma general, la base de dades dissenyada conté les dades següents:

- Dades personals dels alumnes: nom, cognoms, número d'expedient, any d'ingrés al centre, data de naixement, adreça completa, telèfon i nom dels pares per tal de poder-hi contactar quan sigui necessari.

- Dades personals dels professors: nom, cognoms, adreça completa, titulació.

- Dades relatives als centres escolars: nom, adreça completa.

- Dades relatives als cursos que s'hi fan, així com de les assignatures de què consten i dels diferents calendaris escolars.

- Dades relatives a les amonestacions i sancions que es poden aplicar als alumnes: tipus, gravetat.

La base de dades creada és única per a tots els centres educatius, de manera que caldrà definir un sistema de permisos per tal de garantir que cada un d'ells tingui accés exclusivament a les dades que li correspongui. La gestió bàsica de manteniment de la base de dades serà duta a terme per l'administrador del sistema de cada centre escolar. També hi tindran accés els professors pel que fa a la gestió de les amonestacions i les sancions. Així, en iniciar-se l'aplicació, el sistema demanarà un NIF i una contrasenya per comprovar el tipus de privilegis que té l'usuari que hi vol accedir.

Les opcions de manteniment que ofereix la base de dades permeten desenvolupar les funcionalitats següents:

- alta, baixa i/o modificació d'alumnes;
- alta, baixa i/o modificació de professors;
- alta, baixa i/o modificació de cursos;
- alta, baixa i/o modificació d'assignatures;
- alta, baixa i/o modificació de calendaris escolars;
- alta, baixa i/o modificació d'amonestacions i sancions;

A més, és possible consultar i llistar la informació emmagatzemada. De forma concreta, la base de dades té definits els procediments emmagatzemats que es requereixen per obtenir els llistats següents:

- Llistat de totes les amonestacions imposades
- Llistat dels alumnes d'un curs
- Llistat de tots els tipus d'amonestacions i de sancions disponibles
- Llistat de totes les amonestacions i sancions d'un alumne

Així mateix, proporciona informació estadística mitjançant la implementació d'uns procediments que permeten obtenir dades en temps constant 1, és a dir, fent una SELECT sobre un registre d'una taula.

A nivell general, la base de dades compleix amb els requeriments següents:

- Els procediments emmagatzemats disposen d'un paràmetre de sortida anomenat RSP, de tipus *string*, per indicar si l'execució ha finalitzat amb èxit o no.
- Si un procediment finalitza amb èxit, proporciona com a sortida "OK".
- Si un procediment fracassa, la sortida indica el text "Error + tipus d'error".
- Els procediments disposen del tractament d'excepcions.
- Totes les crides a procediments s'emmagatzemen en una taula de *log*, on hi consta el procediment executat i els paràmetres d'entrada i de sortida.
- Es descriu què fa un procediment a alt nivell.
- Es descriu els tipus i els valors possibles de cada paràmetre d'entrada en els procediments.
- Es descriu els tipus i els valors possibles de cada paràmetre de sortida d'un procediment, incloent els diferents errors que pot retornar.

(índex)

3.1. Gestió d'alumnes

Pel que fa a la gestió d'alumnes, s'identifiquen una sèrie de casos d'ús i de regles de negoci que hi són aplicables.

3.1.1. Casos d'ús

Cas d'ús número 1: "Alta alumne"
Resum de la funcionalitat: Afegeix un *alumne* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Precondició: L'*alumne* no existeix a la base de dades.
Postcondició: L'*alumne* resta incorporat a la base de dades.
Procés: L'administrador del sistema de cada centre escolar insereix a la base de dades la informació següent, relativa a un *alumne*: número de document identificatiu, nom, primer cognom, segon cognom, número d'expedient escolar, data de naixement, any

d'ingrés, adreça, codi postal, municipi, comarca, província, telèfon, e-mail, nom del pare i nom de la mare. En acceptar, es validen les dades i, si l'*alumne* no existeix, s'incorpora al sistema. Si ja hi consta, el sistema mostra el missatge d'error "L'alumne ja existeix" i les dades introduïdes no es guarden.

Cas d'ús número 2: "Baixa alumne"
Resum de la funcionalitat: Dóna de baixa (lògica) un *alumne* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar alumne
Precondició: L'*alumne* existeix a la base de dades i està actiu.
Postcondició: L'*alumne* consta donat de baixa a la base de dades.
Procés: L'usuari indica al sistema que vol donar de baixa un alumne. S'engega, llavors, els cas d'ús Cercar alumne. Si l'*alumne* ja ha estat donat de baixa prèviament, el sistema li mostra el missatge d'error "Alumne no actiu"; en tancar el missatge, es torna a engegar el cas d'ús Cercar alumne. Si no hi ha cap error, el sistema mostra les dades de l'*alumne* demanat i, en acceptar, es marca com a *alumne* no actiu, però no s'elimina del sistema.

Cas d'ús número 3: "Modificar alumne"
Resum de la funcionalitat: Modifica les dades d'un *alumne* dels que hi ha actius a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar alumne
Precondició: L'*alumne* existeix a la base de dades i està actiu.
Postcondició: Les dades de l'*alumne* estan modificades.
Procés: L'usuari indica al sistema que vol modificar les dades d'un alumne. S'engega, llavors, el cas d'ús Cercar alumne. Si l'*alumne* ha estat donat de baixa prèviament, el sistema li mostra el missatge d'error "Alumne no actiu" i, en tancar-lo, es torna a engegar el cas d'ús Cercar alumne. Si no hi ha cap error, el sistema mostra les dades de l'*alumne* demanat, totes editables excepte l'identificador del registre, el número de document identificatiu de l'alumne i el número d'expedient escolar, que no es poden modificar. En acceptar, el sistema valida les noves dades i les incorpora.

Cas d'ús número 4: "Cercar alumne"
Resum de la funcionalitat: Es cerca un *alumne* a partir del seu número de document identificatiu.
Actor: **Administrador del sistema del centre escolar, professor**
Casos d'ús relacionats: Baixa alumne, Modificar alumne, Amonestar, Sancionar
Precondició: Cap
Postcondició: S'ha cercat i s'ha trobat l'*alumne*.
Procés: Els sistema demana el número de document identificatiu de l'*alumne* a cercar i l'usuari l'introdueix. Si no es troba cap resultat, el sistema notifica el missatge "El número introduït no existeix". En tancar aquest missatge, es dóna a l'usuari l'opció

d'introduir un número nou. Si no hi ha cap error, el sistema mostra les dades de l'*alumne* demanat.

(índex)

3.1.2. Regles de negoci

Les regles de negoci que ha de complir la base de dades, pel que fa a la gestió dels alumnes, són les següents:

- Tot alumne estudia en un centre escolar i només en un.
- Un alumne només pot estar matriculat d'un únic curs.

(índex)

3.2. Gestió de professors

En el cas de la gestió de professors, s'identifiquen els casos d'ús i les regles de negoci següents:

3.2.1. Casos d'ús

Cas d'ús número 5: "Alta professor"
Resum de la funcionalitat: Afegeix un *professor* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Precondició: El *professor* no existeix a la base de dades.
Postcondició: El *professor* resta incorporat a la base de dades.
Procés: L'administrador del sistema de cada centre escolar insereix a la base de dades la informació següent, relativa a un *professor*: número de document identificatiu, nom, primer cognom, segon cognom, titulació, adreça, codi postal, municipi, comarca, província, telèfon i e-mail. En acceptar, es validen les dades i, si el *professor* no existeix, s'incorpora al sistema. Si ja hi consta, el sistema mostra el missatge d'error "El professor ja existeix" i les dades introduïdes no es guarden.

Cas d'ús número 6: "Baixa professor"
Resum de la funcionalitat: Dóna de baixa (lògica) un *professor* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar professor
Precondició: El *professor* existeix a la base de dades i està actiu.
Postcondició: El *professor* consta donat de baixa a la base de dades.
Procés: L'usuari indica al sistema que vol donar de baixa les dades d'un professor. S'engega, llavors, el cas d'ús Cercar professor. Si el *professor* ja ha estat donat de baixa prèviament, el sistema li mostra el missatge d'error "Professor no actiu"; en tancar el missatge, es torna a engegar el cas d'ús Cercar professor. Si no hi ha cap error, el sistema mostra les dades del professor demanat i, en acceptar, es marca com a *professor* no actiu, però no s'elimina del sistema.

Cas d'ús número 7: "Modificar professor"
Resum de la funcionalitat: Modifica les dades d'un *professor* dels que hi ha actius a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar professor
Precondició: El *professor* existeix a la base de dades i està actiu.
Postcondició: Les dades del *professor* estan modificades.
Procés: L'usuari indica al sistema que vol modificar les dades d'un professor. S'engega, llavors, el cas d'ús Cercar professor. Si el *professor* ha estat donat de baixa prèviament, el sistema li mostra el missatge d'error "Professor no actiu" i, en tancar-lo, es torna a engegar el cas d'ús Cercar professor. Si no hi ha cap error, el sistema mostra les dades del *professor* demanat, totes editables excepte l'identificador del registre i el número de document identificatiu del professor, que no es poden modificar. En acceptar, el sistema valida les noves dades i les incorpora.

Cas d'ús número 8: "Cercar professor"
Resum de la funcionalitat: Es cerca un *professor* a partir del seu número de document identificatiu.
Actor: **Administrador del sistema del centre escolar, professor**
Casos d'ús relacionats: Baixa professor, Modificar professor
Precondició: Cap
Postcondició: S'ha cercat i s'ha trobat el *professor*.
Procés: El sistema demana el número de document identificatiu del *professor* a cercar i l'usuari l'introdueix. Si no es troba cap resultat, el sistema notifica el missatge "El número introduït no existeix". En tancar aquest missatge, es dóna a l'usuari l'opció d'introduir un número nou. Si no hi ha cap error, el sistema mostra les dades del *professor* demanat.

(índex)

3.2.2. Regles de negoci

Pel que fa a les regles de negoci que la base de dades ha de complir en relació amb la gestió de professors, s'especifiquen tot seguit:

- Tot professor treballa en un centre escolar i només en un. En un centre escolar, però, treballen molts professors.
- Un professor pot impartir cap, una o vàries assignatures diferents.
- Un professor pot donar classes a més d'un curs.
- Un professor pot ser tutor d'un o de diversos cursos o també no ser-ne responsable de cap.
- Un alumne només pot tenir un únic professor tutor.
- Un professor tutor té definit un únic dia de consulta als alumnes per cada curs de què es responsable.

- Si un professor és tutor de més d'un curs, els dies de tutoria poden ser els mateixos, però les hores seran diferents per cada un dels cursos.

(índex)

3.3. Gestió de cursos, assignatures i calendaris escolars

Quant a la gestió de cursos, assignatures i calendaris escolars, els casos d'ús i les regles de negoci que s'apliquen s'especifiquen tot seguit.

3.3.1. Casos d'ús

Cas d'ús número 9: "Alta curs"
Resum de la funcionalitat: Afegeix un *curs* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Assignar assignatura a un curs, Crear calendari escolar
Precondició: El *curs* no existeix a la base de dades.
Postcondició: El *curs* resta incorporat a la base de dades.
Procés: L'administrador del sistema de cada centre escolar insereix a la base de dades un nom descriptiu del curs i el nombre de grups que tindrà. En acceptar, es validen les dades i, si el *curs* no existeix, s'incorpora al sistema. Si ja hi consta, el sistema mostra el missatge d'error "El curs ja existeix" i les dades introduïdes no es guarden.

Cas d'ús número 10: "Baixa curs"
Resum de la funcionalitat: Dóna de baixa (lògica) un *curs* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar curs
Precondició: El *curs* existeix a la base de dades i està actiu.
Postcondició: El *curs* consta donat de baixa a la base de dades.
Procés: L'usuari indica al sistema que vol donar de baixa un curs. S'engega, llavors, el cas d'ús Cercar curs. Si el *curs* ja ha estat donat de baixa prèviament, el sistema li mostra el missatge d'error "Curs no actiu"; en tancar el missatge, es torna a engegar el cas d'ús Cercar curs. Si no hi ha cap error, el sistema mostra les dades del *curs* demanat i, en acceptar, es marca com a *curs* no actiu, però no s'elimina del sistema.

Cas d'ús número 11: "Modificar curs"
Resum de la funcionalitat: Modifica les dades d'un *curs* dels que hi ha actius a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar curs
Precondició: El *curs* existeix a la base de dades i està actiu.
Postcondició: Les dades del *curs* estan modificades.
Procés: L'usuari indica al sistema que vol modificar les dades d'un curs. S'engega, llavors, el cas d'ús Cercar curs. Si el *curs* ha estat donat de baixa prèviament, el sistema li mostra el missatge d'error "Curs no actiu" i, en tancar-lo, es torna a engegar el cas

d'ús Cercar curs. Si no hi ha cap error, el sistema mostra les dades del *curs* demanat, però només amb la possibilitat de modificar-ne el nom o el nombre de grups; l'identificador del *curs* no es mostra editable. En acceptar, el sistema valida les noves dades i les incorpora.

Cas d'ús número 12: "Cercar curs"
Resum de la funcionalitat: Es cerca un *curs* a partir del seu identificador.
Actor: **Administrador del sistema del centre escolar, professor**
Casos d'ús relacionats: Baixa curs, Modificar curs, Assignar assignatura a un curs, Crear calendari escolar
Precondició: Cap
Postcondició: S'ha cercat i s'ha trobat el *curs*.
Procés: El sistema demana l'identificador del *curs* a cercar i l'usuari l'introdueix. Si no es troba cap resultat, el sistema notifica el missatge "L'identificador introduït no existeix". En tancar aquest missatge, es dóna a l'usuari l'opció d'introduir un identificador nou. Si no hi ha cap error, el sistema mostra les dades del *curs* demanat.

Cas d'ús número 13: "Alta assignatura"
Resum de la funcionalitat: Afegeix un *assignatura* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Precondició: L'*assignatura* no existeix a la base de dades.
Postcondició: L'*assignatura* resta incorporada a la base de dades.
Procés: L'administrador del sistema de cada centre escolar insereix a la base de dades l'identificador de l'assignatura i un nom descriptiu. En acceptar, es validen les dades i, si l'*assignatura* no existeix, s'incorpora al sistema. Si ja hi consta, el sistema mostra el missatge d'error "L'assignatura ja existeix" i les dades introduïdes no es guarden.

Cas d'ús número 14: "Baixa assignatura"
Resum de la funcionalitat: Dóna de baixa (lògica) una *assignatura* a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar assignatura
Precondició: L'*assignatura* existeix a la base de dades i està activa.
Postcondició: L'*assignatura* consta donada de baixa a la base de dades.
Procés: L'usuari indica al sistema que vol donar de baixa una assignatura. S'engega, llavors, els cas d'ús Cercar assignatura. Si l'*assignatura* ja ha estat donada de baixa prèviament, el sistema li mostra el missatge d'error "Assignatura no activa"; en tancar el missatge, es torna a engegar el cas d'ús Cercar assignatura. Si no hi ha cap error, el sistema mostra les dades de l'assignatura demanada i, en acceptar, es marca com a *assignatura* no activa, però no s'elimina del sistema.

Cas d'ús número 15: "Modificar assignatura"
Resum de la funcionalitat: Modifica les dades d'una *assignatura* de les que hi ha actives a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Casos d'ús relacionats: Cercar assignatura

Precondició: L'*assignatura* existeix a la base de dades i està activa.
Postcondició: Les dades de l'*assignatura* estan modificades.
Procés: L'usuari indica al sistema que vol modificar les dades d'una assignatura. S'engega, llavors, el cas d'ús Cercar assignatura. Si l'*assignatura* ha estat donada de baixa prèviament, el sistema li mostra el missatge d'error "Assignatura no activa" i, en tancar-lo, es torna a engegar el cas d'ús Cercar assignatura. Si no hi ha cap error, el sistema mostra les dades de l'*assignatura* demanada, però només amb la possibilitat de modificar-ne el nom; l'identificador de l'*assignatura* no es mostra editable. En acceptar, el sistema valida les noves dades i les incorpora.

Cas d'ús número 16: "Cercar assignatura"
Resum de la funcionalitat: Es cerca una *assignatura* a partir del seu identificador.
Actor: **Administrador del sistema del centre escolar, professor**
Casos d'ús relacionats: Baixa assignatura, Modificar assignatura
Precondició: Cap
Postcondició: S'ha cercat i s'ha trobat l'*assignatura*.
Procés: El sistema demana l'identificador de l'*assignatura* a cercar i l'usuari l'introdueix. Si no es troba cap resultat, el sistema notifica el missatge "L'identificador introduït no existeix". En tancar aquest missatge, es dóna a l'usuari l'opció d'introduir un identificador nou. Si no hi ha cap error, el sistema mostra les dades de l'*assignatura* demanada.

Cas d'ús número 17: "Assignar assignatura a un curs"
Resum de la funcionalitat: Assigna les *assignatures* a cada un dels *cursos* que hi ha a la base de dades.
Actor: **Administrador del sistema del centre escolar**
Precondició: L'*assignatura* i el *curs* existeixen a la base de dades.
Postcondició: L'*assignatura* queda assignada al curs.
Casos d'ús relacionats: Cercar curs, Alta curs, Alta assignatura
Procés: L'usuari indica al sistema que vol assignar una *assignatura* a un *curs*. S'engega, en primer lloc, el cas d'ús Cercar curs. Si el *curs* no existeix, s'engega el cas d'ús Alta curs. Si existeix, el sistema mostra les dades del *curs* demanat. L'usuari demana la relació d'assignatures existents i el sistema les hi mostra. L'usuari en selecciona una. El sistema registra l'assignació de l'assignatura seleccionada al curs indicat i mostra la relació de les restants per continuar amb la selecció fins que es finalitza el cas d'ús.
Alternatives: Si no hi ha assignatures emmagatzemades, cal engegar el cas d'ús Alta assignatura.

Cas d'ús número 18: "Crear calendari escolar"
Resum de la funcionalitat: Crea un calendari escolar per a un *curs* definit.
Actor: **Administrador del centre escolar**
Precondició: L'*assignatura* i el *curs* existeixen a la base de dades.
Postcondició: El calendari escolar queda definit a la base de dades.
Casos d'ús relacionats: Cercar curs, Alta curs, Assignar assignatura a un curs

Procés: L'usuari indica al sistema que vol crear un calendari escolar per a un *curs*. El sistema li demana, primer, que n'introdueixi un nom i, en fer-ho, engega el cas d'ús Cercar curs. Si el *curs* no existeix, s'engega el cas d'ús Alta curs. Si existeix, el sistema mostra les dades del *curs* demanat amb la relació d'assignatures que té assignades. Si no en té cap d'assignada, s'engega el cas d'ús Assignar assignatura a un curs. L'usuari en selecciona una i el sistema li demana que n'introdueixi l'horari. L'usuari indica el dia de la setmana i l'hora, el sistema enregistra les dades introduïdes i torna a mostrar la relació de les assignatures restants per continuar amb la selecció fins que es finalitza el cas d'ús.

(índex)

3.3.2. Regles de negoci

- Els cursos que s'imparteixen són sis (els quatre cursos d'ESO i els dos de batxillerat) i tots es fan a tots els centres.
- Cada curs està format per diverses assignatures.
- Les assignatures es defineixen de forma genèrica, però s'identifiquen plenament amb el curs a què estan assignades.
- Un mateix curs pot tenir un o més professors responsables.
- Una assignatura pot ser impartida per un o més professors diferents dins del mateix curs.
- El calendari escolar contempla els dies festius.

(índex)

3.4. Gestió d'amonestacions i sancions

Finalment, pel que fa a la gestió d'amonestacions i sancions, s'identifiquen els casos d'ús i les regles de negoci que s'esmenten a continuació.

3.4.1. Casos d'ús

Cas d'ús número 19: "Alta amonestació"
Resum de la funcionalitat: Afegeix una *amonestació* a la base de dades.
Actor: **Administrador del sistema del centre escolar, Professor**
Casos d'ús relacionats: Assignar sanció a amonestació, Amonestar
Precondició: L'*amonestació* no existeix a la base de dades.
Postcondició: L'*amonestació* resta incorporada a la base de dades.
Procés: El professor insereix a la base de dades un tipus d'amonestació, una descripció i un nivell de gravetat. En acceptar, es validen les dades i, si l'*amonestació* no existeix, s'incorpora al sistema. Si ja hi consta, el sistema mostra el missatge d'error "El tipus d'amonestació ja existeix" i les dades introduïdes no es guarden.

Cas d'ús número 20: "Baixa amonestació"
Resum de la funcionalitat: Dóna de baixa (lògica) una *amonestació* a la base de dades.

Actor: **Administrador del sistema del centre escolar, Professor**
Casos d'ús relacionats: Cercar amonestació
Precondició: L'*amonestació* existeix a la base de dades i està activa.
Postcondició: L'*amonestació* consta donada de baixa a la base de dades.
Procés: L'usuari indica al sistema que vol donar de baixa una amonestació. S'engega, llavors, els cas d'ús Cercar amonestació. Si l'*amonestació* ja ha estat donada de baixa prèviament, el sistema li mostra el missatge d'error "Amonestació no activa"; en tancar el missatge, es torna a engegar el cas d'ús Cercar amonestació. Si no hi ha cap error, el sistema mostra les dades de l'*amonestació* demanada i, en acceptar, es marca com a *amonestació* no activa, però no s'elimina del sistema.

Cas d'ús número 21: "Modificar amonestació"
Resum de la funcionalitat: Modifica les dades d'una *amonestació* de les que hi ha actives a la base de dades.
Actor: **Administrador del sistema del centre escolar, Professor**
Casos d'ús relacionats: Cercar amonestació
Precondició: L'*amonestació* existeix a la base de dades i està activa.
Postcondició: Les dades de l'*amonestació* estan modificades.
Procés: L'usuari indica al sistema que vol modificar les dades d'una amonestació. S'engega, llavors, el cas d'ús Cercar amonestació. Si l'*amonestació* ha estat donada de baixa prèviament, el sistema li mostra el missatge d'error "Amonestació no activa" i, en tancar-lo, es torna a engegar el cas d'ús Cercar amonestació. Si no hi ha cap error, el sistema mostra les dades de l'*amonestació* demanada, però només amb la possibilitat de modificar-ne els atributs diferents a l'identificador de l'*amonestació,* que no es mostra editable. En acceptar, el sistema valida les noves dades i les incorpora.

Cas d'ús número 22: "Cercar amonestació"
Resum de la funcionalitat: Es cerca una *amonestació* a partir del seu identificador.
Actor:**Administrador del sistema del centre escolar, Professor**
Casos d'ús relacionats: Baixa amonestació, Modificar amonestació, Assignar sanció a amonestació
Precondició: Cap
Postcondició: S'ha cercat i s'ha trobat l'*amonestació*.
Procés: El sistema demana l'identificador de l'*amonestació* a cercar i l'usuari l'introdueix. Si no es troba cap resultat, el sistema notifica el missatge "L'identificador introduït no existeix". En tancar aquest missatge, es dóna a l'usuari l'opció d'introduir un identificador nou. Si no hi ha cap error, el sistema mostra les dades de l'*amonestació* demanada.

Cas d'ús número 23: "Alta sanció"
Resum de la funcionalitat: Afegeix una *sanció* a la base de dades.
Actor: **Administrador del sistema del centre escolar, Professor**
Precondició: La *sanció* no existeix a la base de dades.
Postcondició: La *sanció* resta incorporada a la base de dades.

Procés: L'usuari insereix a la base de dades un identificador de la sanció i una descripció. En acceptar, es validen les dades i, si la *sanció* no existeix, s'incorpora al sistema. Si ja hi consta, el sistema mostra el missatge d'error "El tipus de sanció ja existeix" i les dades introduïdes no es guarden.

Cas d'ús número 24: "Baixa sanció"
Resum de la funcionalitat: Dóna de baixa (lògica) una *sanció* a la base de dades.
Actor: **Administrador del sistema del centre escolar, Professor**
Casos d'ús relacionats: Cercar sanció
Precondició: La *sanció* existeix a la base de dades i està activa.
Postcondició: La *sanció* consta donada de baixa a la base de dades.
Procés: L'usuari indica al sistema que vol donar de baixa una sanció. S'engega, llavors, els cas d'ús Cercar sanció. Si la *sanció* ja ha estat donada de baixa prèviament, el sistema li mostra el missatge d'error "Sanció no activa"; en tancar el missatge, es torna a engegar el cas d'ús Cercar sanció. Si no hi ha cap error, el sistema mostra les dades de la *sanció* demanada i, en acceptar, es marca com a *sanció* no activa, però no s'elimina del sistema.

Cas d'ús número 25: "Modificar sanció"
Resum de la funcionalitat: Modifica les dades d'una *sanció* de les que hi ha actives a la base de dades.
Actor: **Administrador del sistema del centre escolar, Professor**
Casos d'ús relacionats: Cercar sanció
Precondició: La *sanció* existeix a la base de dades i està activa.
Postcondició: Les dades de la *sanció* estan modificades.
Procés: L'usuari indica al sistema que vol modificar les dades d'una sanció. S'engega, llavors, el cas d'ús Cercar sanció. Si la *sanció* ha estat donada de baixa prèviament, el sistema li mostra el missatge d'error "Sanció no activa" i, en tancar-lo, es torna a engegar el cas d'ús Cercar sanció. Si no hi ha cap error, el sistema mostra les dades de la *sanció* demanada, però només amb la possibilitat de modificar-ne les dades diferents a l'identificador de la *sanció*, que no es mostra editable. En acceptar, el sistema valida les noves dades i les incorpora.

Cas d'ús número 26: "Cercar sanció"
Resum de la funcionalitat: Es cerca una *sanció* a partir del seu identificador.
Actor:**Administrador del sistema del centre escolar, Professor**
Casos d'ús relacionats: Baixa sanció, Modificar sanció
Precondició: Cap
Postcondició: S'ha cercat i s'ha trobat la *sanció*.
Procés: El sistema demana l'identificador de la *sanció* a cercar i l'usuari l'introdueix. Si no es troba cap resultat, el sistema notifica el missatge "L'identificador introduït no existeix". En tancar aquest missatge, es dóna a l'usuari l'opció d'introduir un identificador nou. Si no hi ha cap error, el sistema mostra les dades de la *sanció* demanada.

Cas d'ús número 27: "Assignar sanció a amonestació"
Resum de la funcionalitat: Defineix l'activació de sancions automàtiques, assignant una *sanció* a una *amonestació* de tipus molt greu o comesa de forma reiterada un determinat nombre de vegades.
Actor: **Administrador del sistema del centre escolar, Professor**
Precondició: L'*amonestació* i la *sanció* existeixen a la base de dades.
Postcondició: La *sanció* resta lligada a una o vàries amonestacions definides.
Casos d'ús relacionats: Cercar amonestació**,** Alta amonestació, Alta sanció
Procés: L'usuari indica al sistema que vol assignar una *sanció* a una *amonestació*. S'engega, en primer lloc, el cas d'ús Cercar amonestació. Si l'*amonestació* no existeix, s'engega el cas d'ús Alta amonestació. Si existeix, el sistema mostra les dades de l'*amonestació* demanada. L'usuari demana la relació de *sancions* existents i el sistema les hi mostra. L'usuari en selecciona una. El sistema demana introduir l'operador amb què ha de fer la comparació i el nombre acumulat d'*amonestacions* que s'ha de tenir per activar la *sanció* automàticament. El sistema registra l'assignació de la *sanció* seleccionada a l'*amonestació* i torna a mostrar la relació de les amonestacions restants per si, opcionalment, es vol continuar amb la selecció.
Alternatives: Si la *sanció* desitjada no existeix, s'engega el cas d'ús Alta sanció.

Cas d'ús número 28: "Amonestar"
Resum de la funcionalitat: Dóna d'alta a la base de dades l'*amonestació* a un *alumne*.
Actor: **Professor**
Precondició: L'*amonestació* i l'*alumne* existeixen a la base de dades.
Postcondició: L'*alumne* consta amonestat a la base de dades.
Casos d'ús relacionats: Cercar alumne**,** Alta amonestació
Procés: L'usuari indica al sistema que vol assignar una *amonestació* a un *alumne*. S'engega, en primer lloc, el cas d'ús Cercar alumne. El sistema mostra l'alumne cercat amb indicació del curs en què està matriculat. L'usuari demana les assignatures que cursa *l'alumne* i el sistema les hi mostra. L'usuari en selecciona una. Demana, llavors, les *amonestacions* existents a la base de dades i el sistema li'n presenta la relació. L'usuari en selecciona una. El sistema li demana que afegeixi la data i l'hora i si ho comunica als pares. L'usuari introdueix les dades i el sistema registra l'assignació de l'*amonestació* seleccionada (i la sanció corresponent, si s'escau) a l'*alumne* cercat, afegint les dades relatives al professor que l'ha registrat.
Alternatives: Si l'*amonestació* no existeix, s'engega el cas d'ús Alta amonestació.

Cas d'ús número 29: "Sancionar"
Resum de la funcionalitat: Dóna d'alta a la base de dades una sanció no automàtica a un *alumne*.
Actor: **Professor**
Precondició: La *sanció* i l'*alumne* existeixen a la base de dades.
Postcondició: L'*alumne* consta sancionat a la base de dades.
Casos d'ús relacionats: Cercar alumne**,** Alta sanció

Procés: L'usuari indica al sistema que vol assignar una *sanció* (personalitzada) a un *alumne*. S'engega, en primer lloc, el cas d'ús Cercar alumne. El sistema mostra l'*alumne* cercat amb indicació del curs en què està matriculat. L'usuari demana les *sancions* existents a la base de dades i el sistema li'n presenta la relació. L'usuari en selecciona una. El sistema li demana que afegeixi el motiu pel qual s'activa la sanció i la data. L'usuari introdueix les dades i el sistema registra l'assignació de la *sanció* seleccionada a l'*alumne* cercat, afegint les dades relatives al professor que l'ha registrat.
Alternatives: Si la *sanció* no existeix, s'engega el cas d'ús Alta sanció.

(índex)

3.4.2. Regles de negoci

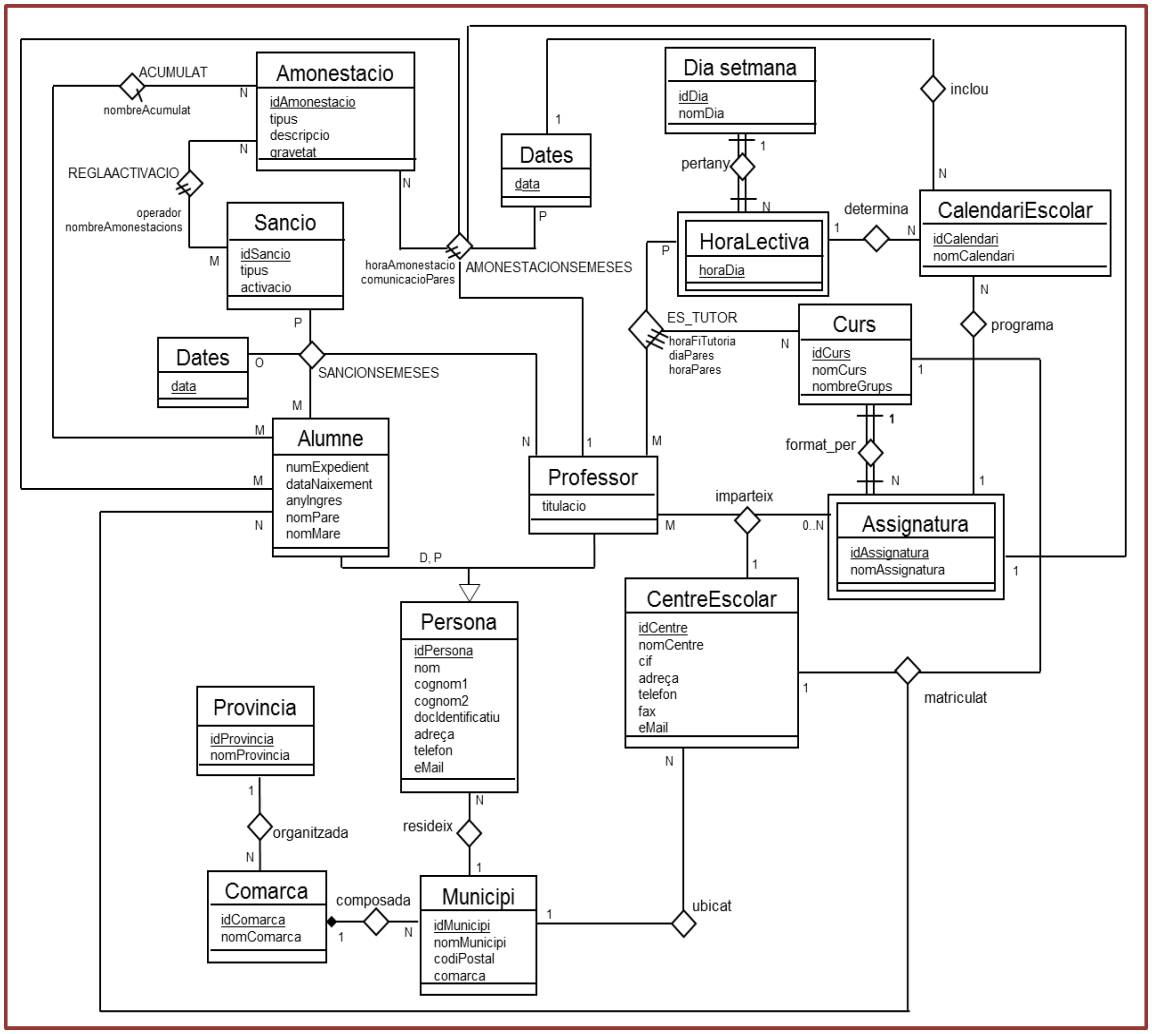
- Una mateixa sanció pot ser aplicada a diversos tipus d'amonestació.
- Una amonestació pot comportar 0, 1 o diversos tipus de sancions.
- Les amonestacions poden ser lleus, greus o molt greus.
- Una amonestació lleu o greu no comporta automàticament una sanció.
- Una amonestació molt greu comporta una sanció.
- L'acumulació d'amonestacions lleus o greus comporten automàticament una sanció, sense que es tingui en compte l'any d'imposició de l'amonestació o els professors que les han imposat. El sistema defineix les regles per activar sancions automàtiques i permet definir-ne de noves.
- Un professor, opcionalment, pot imposar de forma manual una sanció personalitzada.

(índex)

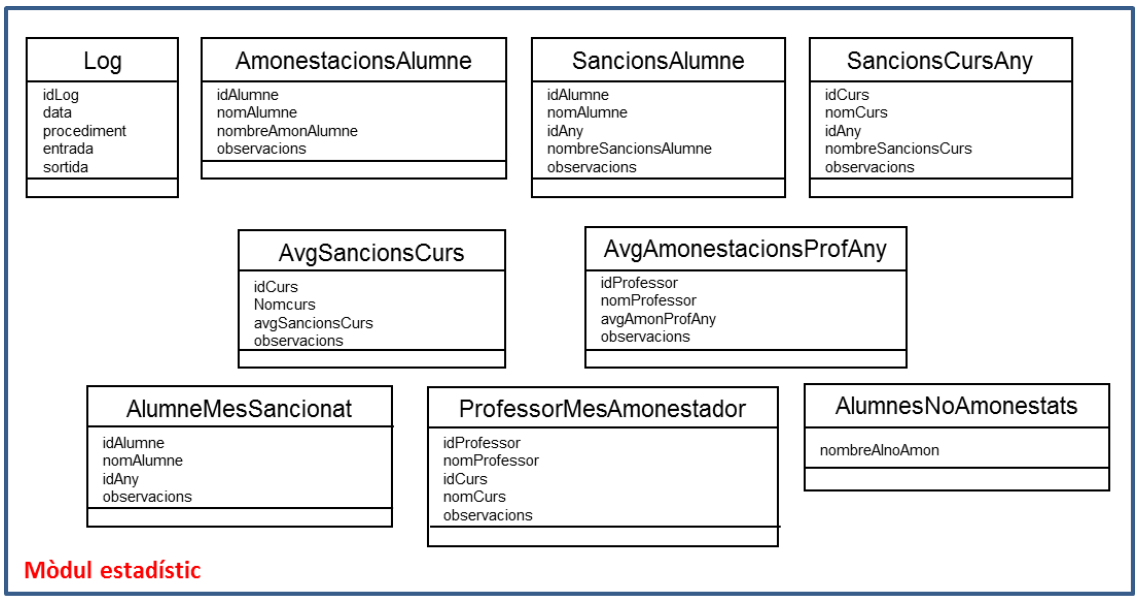
4. DISSENY CONCEPTUAL DE LA BASE DE DADES

4.1. Diagrama E/R

Una vegada coneguts els requeriments i l'especificació dels casos d'ús, s'elabora el diagrama entitat – relació següent:



Nota: S'ha duplicat l'entitat *Dates* per fer més intel·ligible l'esquema.



(índex)

4.2. Identificació d'entitats

D'acord amb els requisits establerts, les entitats que s'identifiquen, amb especificació dels seus atributs, són les següents (clau primària subratllada):

- **Persona**
  idPersona, nom, cognom1, cognom2, docIdentificatiu, adreça, telefon, eMail
- **Alumne** (entitat subclasse de Persona)
  numExpedient, dataNaixement, anyIngres, nomPare, nomMare
- **Professor** (entitat subclasse de Persona)
  titulacio
- **CentreEscolar**
  idCentre, nomCentre, cif, adreça, telefon, fax, eMail
- **Curs**
  idCurs, nomCurs, nombreGrups
- **Assignatura** (entitat dèbil de Curs)
  idAssignatura, nomAssignatura
- **Amonestacio**
  idAmonestacio, tipus, descripcio, gravetat
- **Sancio**
  idSancio, tipus, activacio
- **Dates**
  data, festiu
- **DiaSetmana**
  idDia, nomDia
- **HoraLectiva** (entitat dèbil de DiaSetmana)
  horaDia
- **CalendariEscolar**
  idCalendari, nomCalendari
- **Municipi**
  idMunicipi, nomMunicipi, codiPostal
- **Comarca**
  idComarca, nomComarca
- **Provincia**
  idProvincia, nomProvincia

Justificació:

- A l'entitat genèrica `Persona` es defineix l'atribut *docIdentificatiu* com aquell que guarda qualsevol document que serveix per identificar una persona: DNI, CIF, passaport, targeta de residència, etc.

- S'ha definit l'entitat Assignatura com a entitat dèbil de Curs per considerar que una assignatura necessita del curs en què s'impartirà per quedar completament identificada.
- Igualment, s'ha definit HoraLectiva com a entitat dèbil de DiaSetmana en considerar que les hores escolars queden completament identificades amb el dia de la setmana a què es refereixen. El valor dels dies es restringeix als dies laborables i el de les hores queda acotat entre les 08 i les 18, en considerar-se que aquestes són les hores que abasten l'horari escolar.
- Cada registre de la taula CalendariEscolar representa la programació de les diferents assignatures en les dates corresponents.

(índex)

4.3. Identificació de les interrelacions

Pel que fa a les relacions, les que s'identifiquen són les següents:

Nom: **RESIDEIX**
Entitats relacionades: Municipi, Persona
Connectivitat: 1:N
Descripció: Un determinat municipi defineix el lloc de residència de moltes persones, però una persona només resideix en un únic municipi.

Nom: **UBICAT**
Entitats relacionades: Municipi, Centre Escolar
Connectivitat: 1:N
Descripció: En un determinat municipi hi poden haver diversos centres escolars, però un centre escolar està ubicat en un únic municipi.

Nom: **COMPOSADA**
Entitats relacionades: Comarca, Municipi
Connectivitat: 1:N
Descripció: Tota comarca es compon de diversos municipis, però tot municipi s'adscriu només a una comarca.

Nom: **ORGANITZADA**
Entitats relacionades: Provincia, Comarca
Connectivitat: 1:N
Descripció: Una província es compon de diverses comarques, però una comarca s'adscriu només a una província.

Nom: **FORMAT_PER**
Entitats relacionades: Curs, Assignatura
Connectivitat: 1:N

Descripció: Tot curs està format per diverses assignatures. Una assignatura es defineix pel curs en què s'imparteix.

Nom: **PERTANY**
Entitats relacionades: DiaSetmana, HoraLectiva
Connectivitat: 1:N
Descripció. L'hora lectiva s'identifica parcialment; queda identificada totalment amb el dia de la setmana a què es refereix. Un dia de la setmana té diverses hores lectives.

Nom: **PROGRAMA**
Entitats relacionades: Assignatura, CalendariEscolar
Connectivitat: 1:N
Descripció. Una assignatura queda programada en diversos registres del calendari escolar. Un registre del calendari només fa referència a una determinada assignatura.

Nom: **INCLOU**
Entitats relacionades: Dates, CalendariEscolar
Connectivitat: 1:N
Descripció. Una mateixa data s'inclou en diversos registres del calendari escolar, però un registre del calendari escolar només fa referència a una única data.

Nom: **DETERMINA**
Entitats relacionades: HoraLectiva, CalendariEscolar
Connectivitat: 1:N
Descripció. Una hora lectiva determina diversos registres del calendari escolar, però un registre del calendari només fa referència a una única hora.

Nom: **REGLAACTIVACIO**
Entitats relacionades: Amonestacio, Sancio
Connectivitat: M:N
Descripció: Una amonestació pot implicar diversos tipus de sancions automàtiques, en funció del nombre de reiteracions que es consideri. Una mateixa sanció es pot imposar com a conseqüència de diversos tipus d'amonestacions.

Nom: **ACUMULAT**
Entitats relacionades: Alumne, Amonestacio
Connectivitat: M:N
Descripció: Una alumne pot tenir acumulació de diversos tipus d'amonestació i una determinada amonestació pot estar relacionada de forma reiterada amb diversos alumnes.

Nom: **MATRICULAT**
Entitats relacionades: Alumne, Curs, CentreEscolar
Connectivitat: N:1:1

Descripció: Tot alumne estudia en un únic centre escolar i en un únic curs; en un centre escolar hi estudien molts alumnes que estan matriculats a un únic curs i en un curs d'un centre escolar hi estan matriculats molts alumnes. No pot existir un alumne que no estigui matriculat en cap curs d'un centre escolar ni cap centre escolar o curs que no tingui alumnes matriculats.

Nom: **IMPARTEIX**
Entitats relacionades: Professor, Assignatura, CentreEscolar,
Connectivitat: M:N:1
Descripció: Tot professor treballa en un únic centre escolar i imparteix una o més assignatures; tota assignatura que es fa en un centre escolar és impartida per un o més professors i en un centre escolar hi treballen molts professors que poden donar una o més assignatures. No pot existir cap professor que no estigui associat a un centre escolar o que no imparteixi cap assignatura, així com tampoc pot haver cap centre escolar que no tingui professors o imparteixi assignatures.

Nom: **ES_TUTOR**
Entitats relacionades: Professor, Curs, HoraLectiva
Connectivitat: M:N:P
Descripció: Un professor pot ser tutor de més d'un curs i tenir diverses hores de tutoria. Un curs pot tenir més d'un tutor que tinguin diferents hores de tutoria. En una determinada hora poden passar tutoria diversos professors de diversos cursos.

Nom: **AMONESTACIONSEMESES**
Entitats relacionades: Alumne, Professor, Amonestacio, Data, Assignatura
Connectivitat: M:N:P:1:1
Descripció: Fixat un alumne, un professor, una amonestació i una data i hora, només es pot haver posat en una assignatura. Fixada l'assignatura, el professor i la data i hora, es pot haver amonestat diversos alumnes. Fixada l'assignatura, l'alumne i la data i hora, només un professor pot haver imposat l'amonestació. Durant una assignatura en una data concreta, un alumne pot rebre d'un mateix professor diverses amonestacions. Per últim, un alumne pot rebre diversos tipus d'amonestacions per un professor en una mateixa assignatura en dates diferents.

Nom: **SANCIONS_EMESES**
Entitats relacionades: Alumne, Professor, Sancio, Data
Connectivitat: M:N:P:O
Descripció: Un alumne pot ser sancionat per diversos professors amb diferents tipus de sancions en dates diferents. Un professor pot imposar en diverses dates diferents tipus de sancions a diversos alumnes. Una sanció pot ser imposada a diversos alumnes per professors diferents en dates diverses. Per últim, en una mateixa data poden haver diversos alumnes sancionats per professors diferents amb sancions diverses.

(índex)

5. DISSENY LÒGIC DE LA BASE DE DADES

A partir del resultat del disseny conceptual expressat mitjançant el model ER es fa la transformació a una estructura de dades del model relacional. Així, les entitats descrites originen relacions; els atributs i la clau primària de cada entitat es transformen en els atributs i la clau primària de cada relació.

Pel que fa a les interrelacions, donen lloc bé a claus foranes d'alguna relació ja obtinguda, bé a una nova relació segons el grau i la seva connectivitat. Concretament:

- La transformació de les interrelacions binàries 1:1 i 1:N donen lloc a claus foranes.
- La transformació de les interrelacions binàries M:N i les n-àries es tradueixen en noves relacions.

(índex)

5.1. Transformació d'entitats i d'interrelacions 1:N

D'acord amb el model ER descrit, les relacions resultants són les següents:

- **Persona** (idPersona, nom, cognom1, cognom2, docIdentificatiu, adreça, municipi, telefon, eMail)
  on {municipi} referencia Municipi (idMunicipi)
- **Alumne** (idPersona, numExpedient, dataNaixement, anyIngres, nomPare, nomMare)
  on {idPersona} referencia Persona (idPersona)
- **Professor** (idPersona, titulacio)
  on {idPersona} referencia Persona (idPersona)
- **CentreEscolar** (idCentre, nomCentre, cif, adreça, municipi, telèfon, fax, eMail)
  on {municipi} referencia Municipi (idMunicipi)
- **Curs** (idCurs, nomCurs, nombreGrups)
- **Assignatura** (idCurs, idAssignatura, nomAssignatura)
  on {idCurs} referencia Curs (idCurs)
- **Amonestacio** (idAmonestacio, tipus, descripcio, gravetat)
- **Sancio** (idSancio, tipus, activacio)
- **Dates** (data, festiu)
- **DiaSetmana** (idDia, nomDia)
- **HoraLectiva** (idDia, horaDia)
  on {idDia} referencia DiaSetmana (idDia)
- **CalendariEscolar** (idCalendari, nomCalendari, data, curs, assignatura, dia, hora)
  on {data} referencia Dates (data)

{curs, assignatura} referencia Assignatura (idCurs, idAssignatura),
i {dia, hora} referencia HoraLectiva (idDia, horaDia)

- **Municipi** (idMunicipi, nomMunicipi, codiPostal, comarca)
  on {comarca} referencia Comarca (idComarca)
- **Comarca** (idComarca, nomComarca, provincia)
  on {provincia} referencia Provincia (idProvincia)
- **Provincia** (idProvincia, nomProvincia)

(índex)

5.2. Transformació d'interrelacions M:N i *n*-àries

La transformació de les interrelacions identificades és la següent:

- **ReglaActivacio** (amonestacio, sancio, operador, nombreAmonestacions)
  on {amonestacio} referencia Amonestacio (idAmonestacio)
  i {sancio} referencia Sancio (idSancio)

- **Acumulat** (idAlumne, idAmonestacio, nombreAcumulat)
  on {idAlumne} referencia Alumne (idPersona)
  i {idAmonestacio} referencia Amonestacio (idAmonestacio)

- **Es_tutor** (professor, curs, diaTutoria, horaIniciTutoria, horaFiTutoria, diaPares, horaPares)
  on {professor} referencia Professor (idPersona),
  {curs} referencia Curs(idCurs),
  {diaTutoria, horaIniciTutoria} referencia HoraLectiva (idDia, horaDia)
  i {diaPares, horaPares} referencia HoraLectiva (idDia, horaDia)

- **Matriculat** (alumne, curs, centre)
  on {alumne} referencia Alumne (idPersona),
  {curs} referencia Curs (idCurs)

- **Imparteix** (professor, curs, assignatura, centre)
  on {professor} referencia Professor (idPersona),
  {curs, assignatura} referencia Assignatura (idCurs, idAssignatura)
  i {centre} referencia CentreEscolar (idCentre)

- **AmonestacionsEmeses** (alumne, professor, amonestacio, data, horaAmonestacio, curs, assignatura, comunicacioPares)
  on {alumne} referencia Alumne (idPersona),
  {professor} referencia Professor (idPersona),
  {amonestació} referencia Amonestacio (idAmonestacio)
  {data} referencia Dates (data)

i {curs, assignatura} referencia Assignatura (idCurs, idAssignatura)

- **SancionsEmeses** (alumne, professor, sancio, data)
  on {alumne} referencia Alumne (idPersona),
  {professor} referencia Professor (idPersona),
  {sancio} referencia Sancio (idSancio)
  i {data} referencia Dates (data)

(índex)

6. MÒDUL ESTADÍSTIC

La base de dades ofereix la possibilitat d'obtenir dades estadístiques en temps constant 1. Concretament, l'usuari pot obtenir la informació següent:

- Nombre d'amonestacions per alumne, independentment del curs.
- Nombre de sancions per alumne i any.
- Nombre de sancions per curs i any.
- Mitjana d'amonestacions per professor i any.
- Mitjana de sancions dels alumnes per curs.
- Nom de l'alumne més sancionat en un any donat.
- Nom del professor més amonestador en un curs.
- Nombre d'alumnes que no tenen cap amonestació.

La gestió d'aquesta informació es porta a terme mitjançant la creació de taules específiques per cada estadística desitjada, les quals obtenen les dades a través de procediments emmagatzemats ja definits a la base de dades:

- **AmonestacionsAlumne**
  idAlumne, nomAlumne, nombreAmonAlumne, observacions

- **SancionsAlumne**
  idAlumne, idAany, nomAlumne, nombreSancionsAlumne, observacions

- **SancionsCursAny**
  idCurs, idAny, nomCurs, nombreSancions, observacions

- **AvgSancionsCurs**
  idCurs, nomCurs, avgSancionsCurs, observacions

- **AvgAmonestacionsProfAny**
  idProfessor, nomProfessor, avgAmonProfAny, observacions

- **AlumneMesSancionat**
  idAlumne, idAny, nomAlumne, observacions

- **ProfessorMesAmonestador**
  idProfessor, idCurs, nomProfessor, nomCurs, observacions

- **AlumnesNoAmonestats**
  nombreAlNoAmon

Així mateix, la base de dades disposa d'una taula específica de *log* per emmagatzemar les crides a procediments:

- **Log**
  idLog, data, procediment, entrada, sortida

(índex)

7. IMPLEMENTACIÓ DEL PROJECTE

Per tal de portar a terme aquesta fase del projecte, primer s'ha creat un entorn de treball propi per a aquest TFC amb la definició d'un nou *tablespace* i un nou usuari, al qual s'han atorgat, posteriorment, els permisos corresponents.

```
CREATE TABLESPACE tfc
DATAFILE 'C:\oraclexe\oradata\xe\eso.dbf' SIZE 29 M
AUTOEXTEND ON
MAXSIZE 29 M
ONLINE PERMANENT
EXTENT MANAGEMENT LOCAL AUTOALLOCATE;

CREATE USER user1
IDENTIFIED BY antonia
DEFAULT TABLESPACE tfc
TEMPORARY TABLESPACE temp
QUOTA UNLIMITED ON tfc;

GRANT
CREATE SESSION,
CREATE TABLE,
CREATE MATERIALIZED VIEW,
CREATE TRIGGER,
CREATE SEQUENCE,
CREATE PROCEDURE,
CREATE CLUSTER
TO USER1;
```

La implementació, pròpiament, del projecte, a més de la creació de les taules, ha comportat la definició dels diversos procediments emmagatzemats especificats a l'enunciat, així com la creació d'un mòdul que proporciona una sèrie de dades estadístiques.

Concretament, i pel que fa als procediments, s'han implementat aquells que permeten les funcionalitats següents:

- alta, baixa i/o modificació d'alumnes;
- alta, baixa i/o modificació de professors;
- alta, baixa i/o modificació de cursos;
- alta, baixa i/o modificació d'assignatures;
- alta, baixa i/o modificació de calendaris escolars;
- alta, baixa i/o modificació d'amonestacions i sancions;
- llistar totes les amonestacions imposades
- llistar els alumnes d'un curs
- llistar tots els tipus d'amonestacions i de sancions disponibles
- llistar totes les amonestacions i sancions d'un alumne

A més d'aquests procediments, s'han implementat d'altres no explícitament demanats, però que s'han considerat necessaris per a un correcte desenvolupament del projecte. Així, s'han definit, també, els procediments que s'expliquen tot seguit:

- Alta, baixa i/o modificació de Persones.
  La implementació d'aquest procediment ha estat necessària atesa la definició del model de base de dades que s'ha plantejat, en què els alumnes i els professors s'han dissenyat com a especialitats (subclasses) d'una entitat més genèrica (`Persona`), la qual cosa ha comportat la creació de 3 taules: `Persona` (genèrica), `Alumne` i `Professor`, aquestes dues darreres fent referència a la primera mitjançant la clau forana corresponent.

- Alta de les Amonestacions Emeses i alta de les Sancions Emeses
  S'ha considerat necessari definir aquests dos procediments per tal de facilitar la gestió del mòdul estadístic.

- Procediments per actualitzar específicament cada una de les taules del mòdul estadístic: `estAmonestacionsAlumne`, `estSancionsAlumne`, `estSancionsCursAny`, `estAvgAmonProfAny`, `estAvgSancionsCurs`, `estAlumneMesSancionat`, `estProfMesAmon`, `estAlumnesNoAmon`

Quant a aquest mòdul, l'estadístic, atès que ha d'oferir informació en temps constant 1, s'ha optat per implementar-lo mitjançant la creació d'una taula específica per cada tipus d'estadística demanada, les dades de les quals s'omplen amb la gestió dels procediments emmagatzemats que hi tenen relació.

(índex)

7.1. Taules

En la creació de les taules corresponents s'ha tingut en compte l'especificació de clàusules *NOT NULL* en aquells camps en què s'ha cregut necessària la seva presència i s'han definit algunes clàusules *CHECK* que s'han considerat necessàries o apropiades; així, per exemple, en tractar-se d'un projecte relacionat amb centres educatius de l'àmbit de Catalunya, s'ha restringit el contingut de la taula `Provincia` només a aquelles que componen aquesta comunitat autònoma i, en tractar-se de centres escolars d'ensenyament secundari, els cursos s'han restringit als quatre cursos d'ESO i als dos de batxillerat. Igualment, en tractar-se de jornades escolars, s'han restringit els valors dels dies de la setmana i de les hores del dia a aquells que es consideren propis d'aquests horaris.

En concret, s'han implementat les taules següents (clau primària subratllada):

- Taula Provincia:
  Representa les províncies de Catalunya on s'ubiquen els centres escolar i on resideixen els alumnes i professors.
  Atributs:
    - idProvincia
    - nomProvincia

- Taula Comarca:
  Representa les comarques de Catalunya.
  Atributs:
    - idComarca
    - nomComarca
    - provincia

- Taula Municipi:
  Representa tots els municipis de Catalunya.
  Atributs:
    - idMunicipi
    - nomMunicipi
    - codiPostal
    - comarca

- Taula CentreEscolar:
  Representa els centres educatius que ofereixen els estudis d'ensenyament secundari en l'àmbit territorial de Catalunya.
  Atributs:
    - idCentre
    - nomCentre

- cif (és únic)
- adreça
- municipi
- telefon
- fax
- eMail

❖ Taula Persona:
Recull les dades bàsiques de tota persona, de forma genèrica.
Atributs:

- idPersona
- nom
- cognom1
- cognom2
- docIdentificatiu (DNI, NIF, passaport ...; és únic)
- adreça
- municipi
- telefon
- eMail

❖ Taula Alumne:
Recull dades específiques d'aquelles persones que tenen la condició d'alumne.
Atributs:

- idPersona
- numExpedient (és únic)
- dataNaixement
- anyIngres
- nomPare
- nomMare

❖ Taula Professor:
Recull aquelles dades específiques que corresponen a un professor.
Atributs:

- idPersona
- titulacio

❖ Taula Curs:
Recull les dades que identifiquen un curs.
Atributs:

- idCurs
- nomCurs (és únic)
- nombreGrups

- Taula Assignatura:
  Recull les dades corresponents a les assignatures. Aquestes s'identifiquen en funció del curs a què estan assignades.
  Atributs:
    - idCurs
    - idAssignatura
    - nomAssignatura

- Taula Amonestacio:
  Recull les dades que defineixen els diferents tipus d'amonestació.
  Atributs:
    - idAmonestacio
    - tipus (és únic)
    - descripcio
    - gravetat

- Taula Sancio:
  Recull les dades que defineixen els diferents tipus de sanció.
  Atributs:
    - idSancio
    - tipus (és únic)
    - activacio

- Taula ReglaActivacio:
  Recull aquelles regles que es defineixen per determinar si cal activar una sanció de forma automàtica pel fet de tenir una amonestació molt greu o una acumulació d'amonestacions del mateix tipus.
  Atributs:
    - amonestacio
    - operador
    - nombreAmonestacions
    - sancio

- Taula Dates:
  Relació de dates. Inicialment, la taula recull els dies festius establerts oficialment, així com els caps de setmana i els de vacances dins del curs escolar. La resta de dates s'hi insereixen automàticament en donar d'alta un registre al calendari escolar.
  Atributs:
    - data
    - festiu

- Taula DiaSetmana:
  Recull únicament els dies de la setmana en què poden tenir lloc les classes.
  Atributs:
    - idDia
    - nomDia

- Taula HoraLectiva:
  Representa les hores en què poden tenir lloc les classes. Queden identificades plenament en funció del dia a què facin referència.
  Atributs:
    - idDia
    - horaDia

- Taula CalendariEscolar:
  Recull la programació de les diferents assignatures, amb el seu horari, que tenen lloc durant el curs escolar. Cada registre del calendari correspon a la programació d'una activitat.
  Atributs:
    - idCalendari
    - nomCalendari
    - data
    - curs
    - assignatura
    - dia
    - hora

- Taula Acumulat:
  Taula auxiliar per recollir el nombre d'amonestacions d'un mateix tipus que acumula un alumne per controlar l'activació de sancions automàtiques. El nombre que s'acumula és global, sense tenir en compte l'any en què s'imposa l'amonestació.
  Atributs:
    - idAlumne
    - idAmonestacio
    - nombreAcumulat

- Taula Es_Tutor:
  Representa la interrelació que hi ha entre un professor i el curs del qual és responsable.
  Atributs:
    - professor
    - curs
    - diaTutoria

- horaIniciTutoria
- horaFiTutoria
- diaPares
- horaPares

❖ Taula Matriculat:
Representa la interrelació existent entre un alumne i el curs que estudia.
Atributs:

- alumne
- curs
- centre

❖ Taula Imparteix:
Representa la interrelació existent entre un professor i aquelles assignatures que ensenya.
Atributs:

- professor
- curs
- assignatura
- centre

❖ Taula AmonestacionsEmeses:
Recull les dades bàsiques relatives a la imposició efectiva d'un tipus d'amonestació a un alumne concret per part d'un professor.
Atributs:

- alumne
- professor
- amonestacio
- curs
- assignatura
- data
- horaAmonestacio
- comunicacioPares

❖ Taula SancionsEmeses:
Recull les dades bàsiques relatives a la imposició efectiva d'una sanció a un alumne en concret de forma no automàtica i també aquelles que s'han activat automàticament.
Atributs:

- alumne
- professor
- sancio
- data

A més, pel que fa al mòdul estadístic, s'han creat les taules que s'especifiquen a continuació. En totes elles, excepte en la darrera, s'ha afegit el camp *observacions* per recollir la informació que indica que un determinat alumne, professor o curs han estat donats de baixa.

- Taula AmonestacionsAlumne:
  Aquesta taula proporciona el nombre d'amonestacions per alumne, amb independència del curs a què pertanyen.
  Atributs:
    - idAlumne
    - nomAlumne
    - nombreAmonAlumne
    - observacions

- Taula SancionsAlumne:
  Facilita l'estadística del nombre de sancions que ha rebut cada alumne en cada any. Recull tant les sancions activades automàticament com les imposades de forma personalitzada pels professors.
  Atributs:
    - idAlumne
    - nomAlumne
    - idAny
    - nombreSancionsAlumne
    - observacions

- Taula SancionsCursAny:
  Permet conèixer l'estadística del nombre de sancions que tenen els alumnes per cursos i anys.
  Atributs:
    - idCurs
    - nomCurs
    - idAny
    - nombreSancionsCurs
    - observacions

- Taula AvgAmonestacionsProfAny:
  Proporciona l'estadística de la mitjana d'amonestacions que imposen els diferents professors cada any.
  Atributs:
    - idProfessor
    - nomProfessor
    - avgAmonProfAny
    - observacions

- Taula AvgSancionsCurs:
  Recull l'estadística de la mitjana de sancions que s'han imposat als alumnes per cada curs.
  Atributs:
    - idCurs
    - nomCurs
    - avgSancionsCurs
    - observacions

- Taula AlumneMesSancionat:
  Proporciona el nom de l'alumne que ha rebut més sancions per cada any.
  Atributs:
    - idAlumne
    - nomAlumne
    - idAny
    - observacions

- Taula ProfessorMesAmonestador:
  Facilita el nom del professor que ha posat més amonestacions per cada curs.
  Atributs:
    - idProfessor
    - nomProfessor
    - idCurs
    - nomCurs
    - observacions

- Taula AlumnesNoAmonestats:
  Recull l'estadística del nombre total d'alumnes que no han rebut cap amonestació.
  Atributs:
    - nombreAlNoAmon

Per últim, s'ha implementat la taula *Log* corresponent:

- Taula Log:
  Recull totes les crides als procediments, així com el resultat de l'execució.
  Atributs:
    - idLog
    - data
    - procediment
    - entrada
    - sortida

(S'adjunta, en l'annex, la definició detallada de la implementació de totes les taules, on es mostra de forma explícita les diferents restriccions que se'ls hi aplica, les claus foranes, les claus alternatives, etc.).

(índex)

7.2. Seqüències i disparadors

Per a aquelles taules en què s'ha inclòs un identificador de tipus autonumèric, i per tal que s'incrementin de forma seqüencial, s'han creat les parelles seqüència – disparador corresponents. La seqüència permet incrementar l'identificador cada vegada que s'hi accedeix i el disparador concreta l'esdeveniment pel qual s'hi ha d'incrementar; en aquests casos, abans d'efectuar noves insercions a les taules especificades.

Així, les parelles seqüència – disparador que s'han definit a la base de dades són les següents:

- Taula `Provincia`: Seqüència **seq_Provincia**, Trigger **inserir_idProvincia_Provincia**
- Taula `Comarca`: Seqüència **seq_Comarca**, Trigger **inserir_idComarca_Comarca**
- Taula `Municipi`: Seqüència **seq_Municipi**, Trigger **inserir_idMunicipi_Municipi**
- Taula `Persona`: Seqüència **seq_Persona**, Trigger **inserir_idPersona_Persona**
- Taula `CentreEscolar`: Seqüència **seq_CentreEscolar**, Trigger **inserir_idCentre_Centre**
- Taula `Curs`: Seqüència **seq_Curs**, Trigger **inserir_idCurs_Curs**
- Taula `Amonestacio`: Seqüència **seq_Amon**, Trigger **inserir_idAmon_Amon**
- Taula `Sancio`: Seqüència **seq_Sancio**, Trigger **inserir_idSancio_Sancio**
- Taula `ReglaActivacio`: Seqüència **seq_ReglaActivacio**, Trigger **inserir_idReglaActivacio**
- Taula `DiaSetmana`: Seqüència **seq_DiaSetmana**, Trigger **inserir_idDia_Dia**

- Taula `CalendariEscolar`: Seqüència **seq_Calendari**
  Trigger **inserir_idCalendari**
- Taula `Log`: Seqüència **seq_Log**
  Trigger **inserir_idLog_Log**

No s'han implementat seqüències ni disparadors per a les taules que corresponen al mòdul estadístic atès que es considera que els identificadors seqüencials en aquestes taules no són rellevants ni aporten cap informació.

(Igualment, s'adjunta en l'annex l'especificació detallada de les seqüències i dels disparadors creats).

(índex)

7.3. Funcions

Per tal de facilitar la gestió dels procediments emmagatzemats, s'han definit, també, una sèrie de funcions que ajuden en l'obtenció de les dades que es presenten en les taules del mòdul estadístic, així com unes altres que permeten gestionar l'activació de les sancions automàtiques. En concret, les funcions que s'han implementat són les següents:

- Funció `getNomAlumne`
  L'objectiu d'aquesta funció és retornar el nom d'un alumne
- Funció `getNomProfessor`
  Funció definida per tal d'obtenir el nom d'un professor
- Funció `getNomCurs`
  Funció que retorna el nom d'un curs
- Funció `getOperadorRegla`
  Funció que retorna l'operador definit per les regles d'activació de sancions automàtiques
- Funció `getNombreAcumulatRegla`
  Funció que retorna el nombre d'amonestacions d'un mateix tipus definit en una regla d'activació de sancions automàtiques.
- Funció `getSancioAutomatica`
  Funció que retorna la sanció automàtica definida en una regla d'activació

(Com en els apartats anteriors, s'adjunta en l'annex l'especificació de les funcions creades).

(índex)

7.4. Procediments emmagatzemats

La implementació dels procediments emmagatzemats descrits anteriorment és la part més rellevant d'aquest projecte. En tots ells, en especificar les operacions, es verifica el següent:

- Que no hi ha paràmetres d'entrada nuls que es corresponguin amb aquells atributs que no ho han de ser.
- Que no es creen registres duplicats.
- Que es compleix la unicitat de les claus alternatives.
- Que no es vulnera cap regla de negoci.
- Que s'insereix en la taula `Log` el registre corresponent, indicant el resultat del procediment (paràmetre de sortida RSP).
- Que es gestionen les excepcions, indicant al paràmetre de sortida el tipus d'error que s'ha produït.

El detall dels procediments emmagatzemats que s'han creat es descriu tot seguit. Cal remarcar que llur execució requereix de l'ordre en què s'especifiquen en aquest apartat.

(índex)

7.4.1. Procediments del mòdul estadístic

Per obtenir les dades estadístiques, s'han definit una sèrie de procediments que retornen un valor i que són cridats dins d'altres procediments.

- **`estAmonestacionsAlumne`**: Procediment que gestiona l'actualització dels registres de la taula estadística `AmonestacionsAlumne`, la qual reflecteix el nombre d'amonestacions per alumne, independentment del curs a què pertany.

  - Funcionament:
    a) Comprova que l'identificador de l'alumne que es passa com a paràmetre no sigui nul.
    b) Obté el nom de l'alumne.
    c) Comprova si ja hi ha dades estadístiques relatives a l'alumne.
    d) Si no n'hi ha, hi afegeix la primera amonestació.
    e) Si n'hi ha, obté el nombre d'amonestacions anteriors i li suma l'actual.
    f) Si el procés és correcte, insereix el corresponent registre a la taula `Log`, indicant en el paràmetre de sortida OK.
    g) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_alumne IN Number`: Identificador de l'alumne

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`estSancionsAlumne`**: Procediment que gestiona l'actualització dels registres de la taula estadística `SancionsAlumne`, la qual ofereix el nombre de sancions per alumne i any.

  - Funcionament:
    a) Comprova que els paràmetres que es passen com a entrada no siguin nuls.
    b) Obté el nom de l'alumne mitjançant la funció corresponent.
    c) Obté l'any.
    d) Comprova si ja hi ha dades estadístiques relatives a l'alumne i l'any.
    e) Si no n'hi ha, hi afegeix la primera.
    f) Si n'hi ha, obté el nombre de sancions anteriors i li suma l'actual.
    g) Si el procés és correcte, insereix el corresponent registre a la taula `Log`, indicant en el paràmetre de sortida OK.
    h) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_alumne IN Number`: Identificador de l'alumne
    `p_data IN Date`: Data de la sanció

  - Paràmetre de sortida
    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`estAvgAmonProfAny`**: Procediment que gestiona l'actualització dels registres de la taula estadística `AvgAmonestacionsProfAny`, la qual ofereix la mitjana d'amonestacions per professor i any.

  - Funcionament:
    a) Comprova que els paràmetres d'entrada que es passen no siguin nuls.
    b) Calcula la mitjana requerida.
    c) Obté el nom del professor mitjançant la funció corresponent.
    d) Comprova si ja hi ha dades estadístiques relatives al professor.
    e) Si no n'hi ha, hi afegeix la mitjana obtinguda.
    f) Si n'hi ha, actualitza la dada.
    g) Si el procés és correcte, insereix el corresponent registre a la taula `Log`, indicant en el paràmetre de sortida OK.

a) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_professor IN Number`: Identificador del professor
  `p_data IN Date`: Data de la sanció

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`estSancionsCursAny`**: Procediment que gestiona l'actualització dels registres de la taula estadística `SancionsCursAny`, la qual ofereix el nombre de sancions per curs i any.

  - Funcionament:
    a) Comprova que els paràmetres que es passen com a entrada no siguin nuls.
    b) Obté el nom del curs mitjançant la funció definida a l'efecte.
    c) Obté l'any.
    d) Obté l'identificador del curs.
    e) Comprova si ja hi ha dades estadístiques relatives al curs i l'any.
    f) Si no n'hi ha, hi afegeix la primera.
    g) Si n'hi ha, obté el nombre de sancions anteriors i li suma l'actual.
    h) Si el procés és correcte, insereix el corresponent registre a la taula `Log`, indicant en el paràmetre de sortida OK.
    i) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_alumne IN Number`: Identificador de l'alumne
    `p_data IN Date`: Data de la sanció

  - Paràmetre de sortida
    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`estAlumneMesSancionat`**: Procediment que gestiona l'actualització dels registres de la taula estadística `AlumneMesSancionat`, la qual proporciona el nom de l'alumne que ha rebut més sancions cada any.

- Funcionament:
  a) Comprova que els paràmetres que es passen com a entrada no siguin nuls.
  b) Obté el nom de l'alumne fent servir la funció corresponent.
  c) Comprova si ja hi ha dades estadístiques per a aquell any.
  d) Si no n'hi ha, afegeix el nom de l'alumne obtingut.
  e) Si n'hi ha, en primer lloc, mitjançant un cursor definit prèviament, comprova si hi ha més d'un alumne amb el mateix nombre de sancions màxim. Si n'hi ha més d'un, en la taula mostra el missatge `'No hi ha un únic alumne amb un màxim de sancions'`; si només n'hi ha un, actualitza l'anterior alumne que tenia més sancions, si s'escau.
  f) Si el procés és correcte, insereix el corresponent registre a la taula `Log`, indicant en el paràmetre de sortida, bé OK quan només hi ha un únic alumne més sancionat, bé el mateix text indicatiu de l'existència de més d'un alumne amb un màxim de sancions ja referit anteriorment quan és el cas.
  g) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_alumne IN Number`: Identificador de l'alumne
  `p_data IN Date`: Data de la sanció

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`estProfMesAmon`**: Procediment que gestiona l'actualització dels registres de la taula estadística `ProfessorMesAmonestador`, la qual mostra el nom del professor que ha imposat més amonestacions per cada curs.

  - Funcionament:
    a) Comprova que els paràmetres que es passen com a entrada no siguin nuls.
    b) Obté el nom del professor fent servir la funció corresponent.
    c) Obté el nom del curs mitjançant la funció definida a l'efecte.
    d) Comprova si ja hi ha dades estadístiques per a aquest curs.
    e) Si no n'hi ha, les afegeix.
    f) Si n'hi ha, en primer lloc, mitjançant un cursor definit prèviament, comprova si hi ha més d'un professor amb un mateix nombre d'amonestacions màxim. Si n'hi ha més d'un, en la taula mostra el missatge `'No hi ha un únic professor com a màxim`

amonestador'; si només n'hi ha un, actualitza l'anterior professor que havia imposat més amonestacions, si s'escau.

g) Si el procés és correcte, insereix el corresponent registre a la taula Log, indicant en el paràmetre de sortida, bé OK quan només hi ha un únic professor més amonestador, bé el mateix text indicatiu de l'existència de més d'un ja referit anteriorment si és el cas.
h) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  p_professor IN Number: Identificador del professor
  p_alumne IN Number: Identificador de l'alumne
  p_curs IN Number: Identificador del curs

- Paràmetre de sortida
  RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **estAvgSancionsCurs**: Procediment que gestiona l'actualització dels registres de la taula estadística AvgSancionsCurs, la qual ofereix informació sobre la mitjana de sancions que tenen els alumnes de cada curs.

  - Funcionament:
    a) Comprova que els paràmetres que es passen com a entrada no siguin nuls.
    b) Calcula la mitjana requerida.
    c) Obté el nom del curs mitjançant la funció definida a l'efecte.
    d) Obté l'identificador del curs.
    e) Comprova si ja hi ha dades estadístiques relatives al curs.
    f) Si no n'hi ha, hi afegeix la mitjana obtinguda.
    g) Si n'hi ha, actualitza la dada.
    h) Si el procés és correcte, insereix el corresponent registre a la taula Log, indicant en el paràmetre de sortida OK.
    i) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    p_alumne IN Number: Identificador de l'alumne

  - Paràmetre de sortida
    RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **estAlumnesNoAmon**: Procediment que gestiona l'actualització del registre de la taula estadística `AlumnesNoAmonestats`, la qual mostra el nombre total d'alumnes que no han rebut cap amonestació, sense fer cap tipus de filtrat.

  - Funcionament:
    a) Calcula el nombre d'alumnes no amonestats.
    b) Actualitza la dada a la taula estadística.
    c) Si el procés és correcte, insereix el corresponent registre a la taula `Log`, indicant en el paràmetre de sortida OK.
    d) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    Aquest procediment no té cap paràmetre d'entrada.

  - Paràmetre de sortida
    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

7.4.2. Procediments d'alta

- **altaPersona**: Aquest procediment permet afegir una nova persona a la base de dades global. En aquest procediment no es té en compte el centre a què s'adscriurà. És mitjançant les insercions a les taules *Matriculat* i *Imparteix* que els alumnes i els professors, respectivament, quedaran vinculats amb un centre concret. Els usuaris de cada centre només tindran permisos per accedir a les dades de les persones que hi estiguin vinculades.

  - Funcionament:
    a) Comprova que els paràmetres que no poden ser nuls no ho siguin.
    b) Verifica que el document identificatiu no estigui duplicat.
    c) Constata que el municipi indicat existeix.
    d) Insereix el nou registre a la taula `Persona`.
    e) Si el procés s'executa correctament, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    f) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_nom IN Varchar2`: Nom de la persona
    `p_cognom1 IN Varchar2`: Primer cognom de la persona
    `p_cognom2 IN Varchar2`: Segon cognom de la persona. Pot ser nul

p_docIdentificatiu IN Varchar2: Document que identifica la persona. Pot ser DNI, NIF, targeta de residència, passaport, etc.
p_adreça IN Varchar2: Adreça de la persona (carrer, número, pis...)
p_municipi IN Number: Municipi de residència de la persona.
p_telefon IN Number: Número de telèfon de la persona
p_email IN Varchar2: Correu electrònic de la persona. Pot ser nul.

- Paràmetre de sortida
  RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **altaAlumne**: Aquest procediment permet donar d'alta com a alumne una persona que s'hagi introduït a la base de dades.

  - Funcionament:
    a) Comprova que els paràmetres que no poden ser nuls no ho siguin.
    b) Verifica que el número d'expedient escolar que s'assigni no estigui duplicat.
    c) Constata que la persona indicada s'hagi donat d'alta a la taula Persona.
    d) Verifica que la persona indicada no hagi estat donada d'alta també com a professor.
    e) Si el procés s'executa correctament, insereix el nou registre a la taula Alumne
    f) Actualitza l'estadística del nombre d'amonestacions per alumne, primer cercant el nom de l'alumne mitjançant la funció corresponent, per després afegir aquest nou registre a la taula AmonestacionsAlumne, amb el nombre d'amonestacions a 0.
    g) Actualitza l'estadística del nombre de sancions per alumne i any. Primer, mitjançant un cursor definit prèviament, es busquen els anys, per després inserir, per cada un d'ells, el registre de l'alumne a la taula SancionsAlumne, amb el nombre de sancions a 0.
    h) Actualitza l'estadística del nombre d'alumnes que no tenen cap amonestació, afegint un 1 a la taula AlumnesNoAmonestats si no hi havia dades o sumant 1 al nombre que ja hi hagués.
    i) Si el procés és correcte, actualitza la taula Log, indicant en el paràmetre de sortida OK.
    j) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_idPersona IN Number`: Identificador de la persona que es vol incloure com a alumne
  `p_numExpedient IN Varchar2`: Número d'expedient escolar. No pot ser nul i no es pot repetir.
  `p_dataNaixement IN Date`: Data de naixement de l'alumne.
  `p_anyIngres IN Char`: Any d'ingrés al centre escolar.
  `p_nomPare IN Varchar2`:Nom del pare de l'alumne.
  `p_nomMare IN Varchar2`: Nom de la mare de l'alumne.

- Paràmetre de sortida
  `RSP OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaProfessor`**: Aquest procediment permet donar d'alta com a professor una persona que s'hagi introduït a la base de dades.

  - Funcionament:
    a) Comprova que no siguin nuls aquells paràmetres que no ho han de ser.
    b) Constata que la persona indicada s'hagi donat d'alta a la taula Persona.
    c) Verifica que la persona de referencia no hagi estat donada d'alta com a alumne.
    d) Insereix el nou registre a la taula `Professor`.
    e) Actualitza l'estadística de la mitjana d'amonestacions per professor i any, obtenint, en primer lloc, el nom del professor mitjançant la funció adient i inserint-lo, després, a la taula `AvgAmonestacionsProfAny` amb la dada corresponent a la mitjana a 0.
    f) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    g) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_idPersona IN Number`: Identificador de la persona que es vol incloure com a professor
    `p_titulacio IN Varchar2`: Títol acadèmic que acredita el professor.

  - Paràmetre de sortida
    `RSP OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaCurs`**: Procediment mitjançant el qual es dóna d'alta un curs a la base de dades.

  - Funcionament:
    a) Comprova que no siguin nuls aquells paràmetres que no ho han de ser.
    b) Verifica que el curs no està duplicat.
    c) Controla que no s'introdueixin valors incorrectes per al nom.
    d) Insereix el nou registre a la taula `Curs`.
    e) Actualitza l'estadística de la mitjana de sancions dels alumnes per cursos, obtenint l'identificador del curs i inserint el nou registre a la taula `AvgSancionsCurs`, amb la dada de la mitjana a 0.
    f) Actualitza l'estadística del nombre de sancions per curs i any. Primer, mitjançant un cursor definit prèviament, es busquen els anys, per després inserir, per cada un d'ells, el registre del curs a la taula `SancionsCursAny`, amb el nombre de sancions a 0.
    g) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    h) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_nomCurs IN Varchar2`: Nom del curs que es vol donar d'alta. Ha de complir les restriccions imposades en la definició de la taula.
    `p_nombreGrups IN Number`: Nombre de grups que hi pot haver per cada un dels cursos.

  - Paràmetre de sortida
    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaAssignatura`**: Procediment pel qual es dóna d'alta una assignatura a la base de dades.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Verifica que existeixi el curs a què s'ha d'assignar l'assignatura.
    c) Verifica que l'assignatura no es dupliqui.
    d) Insereix el nou registre a la taula `Assignatura`.
    e) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.

f) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada

  `p_idCurs IN Number`: Identificador del curs a què s'assigna l'assignatura.

  `p_idAssignatura IN Number`: Identificador de l'assignatura.

  `p_nomAssignatura IN Varchar2`: Nom de l'assignatura.

- Paràmetre de sortida

  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaAmonestacio`**: Procediment que gestiona l'alta dels tipus d'amonestació a la base de dades.

  - Funcionament:

    a) Comprova que no siguin nuls aquells paràmetres que no ho han de ser.

    b) Verifica que s'introdueixin valors correctes en el camp gravetat.

    c) Verifica que el tipus d'amonestació no es dupliqui.

    d) Insereix el nou registre a la taula `Amonestacio`.

    e) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.

    f) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada

    `p_tipus IN Varchar2`: Tipus d'amonestació. Ha de ser únic.

    `p_descripcio IN Varchar2`: Descriu quan s'imposa aquest tipus d'amonestació.

    `p_gravetat IN Varchar2`: N'indica la gravetat. Només pot ser lleu, greu o molt greu.

  - Paràmetre de sortida

    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaSancio`**: Procediment que gestiona l'alta dels tipus de sanció a la base de dades.

  - Funcionament:

    a) Comprova que no siguin nuls aquells paràmetres que no ho han de ser.

b) Verifica que el tipus de sanció no es dupliqui.
c) Insereix el nou registre a la taula `Sancio`.
d) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
e) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_tipus IN Varchar2`: Tipus de sanció. Ha de ser únic.
  `p_activacio IN Varchar2`: Descriu quan s'imposa aquest tipus de sanció.

- Paràmetre de sortida
  `RSP OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaCalendariEscolar`**: Procediment mitjançant el qual es donen d'alta a la base de dades els diferents registres d'un calendari escolar, permetent programar les diverses activitats que es fan en cada data pel que fa als diferents cursos i assignatures, establint els horaris d'aquestes.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Verifica que el dia indicat no sigui festiu.
    c) Constata que s'ha donat d'alta l'assignatura i, en conseqüència, existeix el curs.
    d) Comprova que s'ha donat d'alta l'hora lectiva del dia assenyalat.
    e) Verifica que el nou registre no estigui duplicat.
    f) Insereix la data indicada a la taula `Dates` si prèviament no s'havia donat d'alta.
    g) Insereix el nou registre a la taula `CalendariEscolar`.
    h) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    i) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_nomCalendari IN Varchar2`: Identifica el registre del calendari.
    `p_data IN Date`: Data que s'insereix al calendari.
    `p_curs IN Number`: Identificador del curs.
    `p_assignatura IN Number`: Identificador de l'assignatura que es programa.

`p_dia IN Number`: Identificador del dia de la setmana
`p_hora IN Number`: Identificador de l'hora lectiva.

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaAmonestacionsEmeses`**: Procediment que permet donar d'alta a la base de dades una amonestació concreta associada a un alumne. Mitjançant aquest procediment s'actualitzen les dades del mòdul estadístic. Aquest procediment també controla si cal activar una sanció de forma automàtica per acumulació d'amonestacions. Si aquest és el cas, la sanció es comptabilitza en l'any de la darrera amonestació que fa activar-la automàticament.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Verifica que existeix l'alumne a qui s'imposa l'amonestació.
    c) Verifica que existeix el professor que imposa l'amonestació.
    d) Comprova que existeix el tipus d'amonestació que es vol imposar.
    e) Es verifica que el professor amonestador imparteix l'assignatura.
    f) Verifica que el dia indicat no és festiu.
    g) Verifica que el registre resultant no estigui duplicat.
    h) Insereix el nou registre a la taula AmonestacionsEmeses
    i) Calcula si cal activar una sanció automàtica:
       i. Comprova si l'amonestació imposada correspon a alguna de les definides per activar automàticament una sanció.
       ii. Comprova el nombre acumulat d'amonestacions d'aquest tipus que té l'alumne.
       iii. Si no en té cap, li afegeix una.
       iv. Si en té, li suma una altra.
       v. Obté les dades que defineixen les regles d'activació automàtiques mitjançant les funcions definides a l'efecte.
       vi. Comprova si es donen les condicions per a inserir una sanció automàtica.
       vii. Si s'hi donen, insereix una nova sanció a la taula Sancio, posa el comptador d'amonestacions d'aquest tipus a 0 i, en conseqüència, elimina aquest registre de la taula `Acumulat`.
       viii. Actualitza les dades estadístiques relatives a les sancions que es puguin veure afectades.
    j) Actualitza l'estadística del nombre d'amonestacions per alumne mitjançant la crida al procediment corresponent.

k) Actualitza l'estadística del nombre d'alumnes que no tenen cap amonestació amb la crida al procediment definit a l'efecte.
l) Actualitza l'estadística de la mitjana d'amonestacions dels professors per anys per mitjà de la crida al procediment adient.
m) Actualitza l'estadística del nom del professor més amonestador per curs mitjançant la crida al procediment que ho calcula.
n) Si el procés és correcte, actualitza la taula Log, indicant en el paràmetre de sortida OK.
o) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada

  `p_alumne IN Number`: Identificador de l'alumne a qui s'amonesta.
  `p_professor IN Number`: Identificador del professor que amonesta.
  `p_amonestacio IN Number`: Identificador de l'amonestació que s'imposa.
  `p_curs IN Number`: Identificador del curs a què pertany l'alumne.
  `p_assignatura IN Number`: Identificador de l'assignatura en què s'imposa l'amonestació
  `p_data IN Date`: Identificador del dia en què l'alumne és amonestat.
  `p_horaAmonestacio IN Char`: Hora en què s'imposa l'amonestació.
  `p_comunicacioPares IN Char`: Indica si s'ha comunicat als pares.

- Paràmetre de sortida

  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`altaSancionsEmeses`**: Procediment que permet donar d'alta a la base de dades una sanció concreta associada a un alumne quan no s'insereix de forma automàtica. Mitjançant aquest procediment s'actualitzen les dades del mòdul estadístic.

  - Funcionament:

    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Verifica que existeix l'alumne a qui s'imposa la sanció.
    c) Verifica que existeix el professor que imposa la sanció.
    d) Comprova que existeix el tipus de sanció que es vol imposar.
    e) Verifica que el dia indicat no és festiu.
    f) Verifica que el registre resultant no estigui duplicat.
    g) Insereix el nou registre a la taula SancionsEmeses.
    h) Actualitza l'estadística del nombre de sancions per alumne i any amb la crida al procediment corresponent.
    i) Actualitza l'estadística del nombre de sancions per curs i any per mitjà del procediment definit a l'efecte.

j) Actualitza l'estadística de la mitjana de sancions que tenen els alumnes per curs mitjançant la crida al procediment adient.
k) Actualitza l'estadística del nom de l'alumne més sancionat en un any donat amb la crida al procediment que ho calcula.
l) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
m) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_alumne IN Number`: Identificador de l'alumne que és sancionat.
  `p_professor IN Number`: Identificador del professor que sanciona.
  `p_sancio IN Number`: Identificador de la sanció que s'imposa.
  `p_data IN Date`: Identificador del dia en què l'alumne és sancionat.

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

7.4.3. Procediments de baixa

Els procediments de baixa comporten la baixa del registre indicat de la taula a què correspongui. Es produeix, també, la baixa en aquelles taules on estigui referenciat. Els registres, però, no s'eliminen d'aquelles taules que continguin les dades històriques de la base de dades (no implementades en aquest projecte). A més, la baixa no comporta tampoc la modificació de les estadístiques, ja que es considera que aquestes han de perdurar precisament pel seu caràcter estadístic. Així, per exemple, el fet que un alumne o un professor es donin de baixa en un centre no implica que les amonestacions i/o sancions que hagin rebut o imposat, respectivament, no hagin existit, o que modifiqui la mitjana de sancions i/o amonestacions que s'hagin imposat. En aquests casos, s'ha optat per afegir un text indicatiu que l'alumne o professor han estat donats de baixa, però mantenint les seves estadístiques. Sí s'actualitza, però, el nombre d'alumnes no amonestats, atès que sí es considera sense sentit mantenir en aquesta estadística un alumne que ja no hi és.

- **`baixaPersona`:** Mitjançant aquest procediment es gestiona la baixa d'una persona de la taula `Persona`, tant si és alumne com si és professor, ja que la baixa es fa per mitjà del document identificatiu.

  - Funcionament:
    a) Comprova que existeix la persona a qui identifica el document.
    b) Comprova si és alumne o professor.
    c) Si és professor:

i. Dóna de baixa la persona de les taules `AmonestacionsEmeses`, `SancionsEmeses`, `Es_tutor`, `Imparteix` i `Professor`.
ii. Actualitza les diverses taules estadístiques relatives a les amonestacions que han imposat els professors, afegint a totes elles una observació indicativa de que aquest professor en concret ha estat donat de baixa.

d) Si és alumne:
i. Dóna de baixa la persona de les taules `Matriculat`, `AmonestacionsEmeses`, `SancionsEmeses`, `Acumulat` i `Alumne`.
ii. Actualitza les diverses taules estadístiques relatives a les amonestacions rebudes pels alumnes, afegint a totes elles una observació indicativa de que aquest alumne en concret ha estat donat de baixa.

e) Elimina el registre de la taula `Persona`.
f) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
g) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_docIdentificatiu IN Varchar2`: Document que identifica la persona que es vol donar de baixa.

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`baixaCurs`:** Procediment que permet donar de baixa un curs a la base de dades a partir del seu identificador.

  - Funcionament:
    a) Comprova que el curs existeix.
    b) Elimina el registre de totes les taules en què estigui referenciat: `AmonestacionsEmeses`, `Es_tutor`, `Matriculat`, `Imparteix`, `CalendariEscolar` i `Assignatura`
    c) Actualitza les diverses taules estadístiques que fan referència als cursos, afegint a totes elles una observació indicativa de que aquest curs en concret ha estat donat de baixa.
    d) Elimina el registre de la taula `Curs`.
    e) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.

f) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_idCurs IN Number`: Identificador del curs que es dóna de baixa.

- Paràmetre de sortida
  `RSP OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

`baixaAssignatura`: Aquest procediment permet donar de baixa una assignatura a la base de dades a partir dels seus identificadors.

- Funcionament:
  a) Verifica que els identificadors no siguin nuls.
  b) Comprova que l'assignatura existeix.
  c) Elimina el registre de totes les taules en què està referenciat: `AmonestacionsEmeses, Imparteix i CalendariEscolar.`
  d) Elimina el registre de la taula `Assignatura`.
  e) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
  f) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_idCurs IN Number`: Identificador del curs a què està assignada l'assignatura que es dóna de baixa.
  `p_idAssignatura IN Number`: Identificador de l'assignatura que es dóna de baixa.

- Paràmetre de sortida
  `RSP OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

`baixaAmonestacio`: Procediment per donar de baixa una amonestació a la base de dades a partir del seu identificador.

- Funcionament:
  a) Comprova que l'amonestació existeix.
  b) Elimina el registre de totes les taules en què està referenciat: `AmonestacionsEmeses, Acumulat i ReglaActivacio.`
  c) Elimina el tipus d'amonestació de la taula `Amonestacio`.

d) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
e) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_idAmonestacio IN Number`: Identificador de l'amonestació que es dóna de baixa.

- Paràmetre de sortida
  `RSP OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`baixaSancio`:** Procediment per donar de baixa una sanció a la base de dades a partir del seu identificador.

  - Funcionament:
    a) Comprova que la sanció existeix.
    b) Elimina el registre de totes les taules en què està referenciat: `SancionsEmeses i ReglaActivacio.`
    c) Elimina el tipus d'amonestació de la taula `Sancio`.
    d) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    e) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_idSancio IN Number`: Identificador de la sanció que es dóna de baixa.

  - Paràmetre de sortida
    `RSP OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`baixaCalendariEscolar`:** Procediment per donar de baixa un registre del calendari escolar a la base de dades a partir dels seus identificadors.

  - Funcionament:
    a) Comprova que el registre que es vol donar de baixa existeix.
    b) Elimina el registre de la taula `CalendariEscolar`.
    c) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.

d) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_idCalendari IN Number`: Identificador del registre del calendari escolar que es vol donar de baixa.

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

7.4.4. Procediments de modificació

En el cas dels procediments de modificació, comporten canvis en les estadístiques aquells que afecten el nom dels alumnes, professors o cursos. Una modificació en el tipus d'amonestació o de sanció no fa modificar les que anteriorment ja s'haguessin pogut imposar.

- **`modifPersona`:** Procediment per modificar les dades que es refereixen a una persona.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Comprova que la persona existeix.
    c) Verifica que si es modifica el municipi, aquest existeixi.
    d) Es modifiquen les dades que interessen a la taula `Persona`. No es pot modificar, però, el document identificatiu.
    e) Comprova si la persona és alumne o professor per tal d'actualitzar l'estadística corresponent, si s'escau.
    f) Si la persona és alumne i s'ha modificat el nom:
       i. Modifica el nom de l'alumne a la taula `AmonestacionsAlumne`.
       ii. Modifica el nom de l'alumne a la taula `SancionsAlumne`.
       iii. Modifica el nom de l'alumne a la taula `AlumneMesSancionat`, si s'escau.
    g) Si la persona és professor i s'ha modificat el nom:
       i. Actualitza el nom del professor a la taula `ProfessorMesAmonestador`, si s'escau.
       ii. Actualitza el nom del professor a la taula `AvgAmonestacionsProfAny`.
    h) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    i) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  p_nom IN Varchar2: Nom de la persona
  p_cognom1 IN Varchar2: Primer cognom de la persona
  p_cognom2 IN Varchar2: Segon cognom de la persona. Pot ser nul
  p_docIdentificatiu IN Varchar2: Qualsevol document que identifica la persona.
  p_adreça IN Varchar2:Adreça de la persona (carrer, número, pis...)
  p_municipi IN Number: Municipi on resideix la persona.
  p_telefon IN Number: Número de telèfon de la persona
  p_email IN Varchar2: Correu electrònic de la persona. Pot ser nul.

- Paràmetre de sortida
  RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **modifAlumne**: Procediment per modificar les dades que es refereixen a un alumne. No es permet modificar el número d'expedient escolar.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Comprova que l'alumne existeix.
    c) Modifica les dades que interessen a la taula Alumne.
    d) Si el procés és correcte, actualitza la taula Log, indicant en el paràmetre de sortida OK.
    e) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    p_idPersona IN Number: Identificador de l'alumne.
    p_dataNaixement IN Date: Data de naixement de l'alumne.
    p_anyIngres IN Char: Any d'ingrés al centre escolar.
    p_nomPare IN Varchar2:Nom del pare de l'alumne.
    p_nomMare IN Varchar2: Nom de la mare de l'alumne.

  - Paràmetre de sortida
    RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **modifProfessor**: Procediment per modificar les dades que es refereixen a un professor.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.

b) Comprova que el professor existeix.
c) Modifica les dades a la taula `Professor`, excepte l'identificador.
d) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
e) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_idPersona IN Number`: Identificador del professor. No es pot modificar.
  `p_titulacio IN Varchar2`: Títol acadèmic del professor.

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`modifCurs`**: Procediment per modificar les dades relatives a un curs.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Comprova que el curs existeix.
    c) Verifica que el curs que es modifiqui no estigui duplicat.
    d) Verifica que s'introdueixi un valor correcte en el nom del curs.
    e) Modifica les dades a la taula `Curs`, excepte l'identificador.
    f) Si s'ha modificat el nom del curs:
       i. Modifica el curs a la taula `SancionsCursAny`.
       ii. Modifica el curs a la taula `AvgSancionsCurs`.
    g) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    h) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_idCurs IN Number`: Identificador del curs a modificar.
    `p_nomCurs IN Varchar2`: Nom del curs que es vol modificar.
    `p_nombreGrups IN Number`: Nombre de grups que pot tenir el curs.

  - Paràmetre de sortida
    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat.

(índex)

- **`modifAssignatura`**: Procediment per modificar les dades que es refereixen a una assignatura.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Comprova que l'assignatura existeix.
    c) Verifica que l'assignatura resultant de la modificació no estigui duplicada.
    d) Modifica les dades a la taula `Assignatura`, excepte els identificadors.
    e) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    f) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_idCurs IN Number`: Identificador del curs a què s'assigna l'assignatura.
    `p_idAssignatura IN Number`: Identificador de l'assignatura.
    `p_nomAssignatura IN Varchar2`: Nom de l'assignatura.

  - Paràmetre de sortida
    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`modifAmonestacio`**: Procediment per modificar les dades que es refereixen a una amonestació.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Comprova que l'amonestació existeix.
    c) Verifica que, si es modifica, el tipus d'amonestació no es dupliqui.
    d) Verifica que, si es modifica el camp gravetat, s'introdueixin nous valors correctes.
    e) Modifica les dades a la taula `Amonestacio`, excepte l'identificador.
    f) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    g) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_idAmonestacio IN Number`: Identificador de l'amonestació.
    `p_tipus IN Varchar2`: Tipus d'amonestació. Ha de ser únic.
    `p_descripcio IN Varchar2`: Descriu quan s'imposa aquest tipus d'amonestació.

p_gravetat IN Varchar2: N'indica la gravetat, que pot ser lleu, greu o molt greu.

- Paràmetre de sortida

  RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **modifSancio**: Procediment per modificar les dades que es refereixen a una sanció

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Comprova que la sanció existeix.
    c) Verifica que, si es modifica, el tipus de sanció no es dupliqui.
    d) Modifica les dades a la taula Sancio, excepte l'identificador.
    e) Si el procés és correcte, actualitza la taula Log, indicant en el paràmetre de sortida OK.
    f) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada

    p_idSancio IN Number: Identificador de la sanció.

    p_tipus IN Varchar2: Tipus de sanció. Ha de ser únic.

    p_activacio IN Varchar2: Descriu quan s'imposa aquest tipus de sanció.

  - Paràmetre de sortida

    RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **modifCalendariEscolar**: Procediment que permet modificar les dades que es refereixen a una activitat programada en el calendari escolar.

  - Funcionament:
    a) Comprova que els paràmetres que no han de ser nuls no ho siguin.
    b) Verifica que existeix el registre del calendari escolar que es modifica.
    c) Verifica que, si es modifica la data, la nova data no sigui un dia festiu.
    d) Comprova que, si es modifica l'assignatura, aquesta existeixi.
    e) Comprova que, si es modifica l'hora lectiva, aquesta existeixi.
    f) Verifica que el registre resultant de la modificació no estigui duplicat.
    g) Si no hi és, insereix la nova data a la taula Dates..

h) Modifica les dades a la taula `CalendariEscolar`, excepte l'identificador.
i) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
j) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada
  `p_idCalendari IN Number`: Identificador del registre del calendari que es vol modificar.
  `p_nomCalendari IN Varchar2`: Nom del calendari escolar.
  `p_data IN Date`: Data que de l'activitat programada al calendari.
  `p_curs IN Number`: Identificador del curs.
  `p_assignatura IN Number`: Identificador de l'assignatura.
  `p_dia IN Number`: Identificador del dia de la setmana
  `p_hora IN Number`: Identificador de l'hora lectiva.

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

7.4.5. Procediments de consulta (llistats)

Pel que fa als procediments de consultes, permeten obtenir llistats amb la informació actualitzada al moment de la consulta. S'han implementat els següents:

- **`llistatAmonestacionsImposades`**: Procediment de consulta que permet obtenir un llistat de totes les amonestacions imposades, indicant-ne la seva informació bàsica.

  - Funcionament:
    a) Defineix, mitjançant un cursor, la informació que es vol obtenir.
    b) Indica el tipus de llistat (ROWTYPE).
    c) Crea la capçalera del llistat.
    d) Mitjançant un bucle, obté cada una de les files resultants de la consulta definida en el cursor.
    e) Obté el nombre total de registres retornats.
    f) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    g) Mostra el resultat de la consulta.
    h) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    Aquest procediment no té cap paràmetre d'entrada.

- Paràmetre de sortida
  RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`llistatAlumnesCurs`**: Procediment de consulta que permet obtenir un llistat de tots els alumnes d'un curs, indicant-ne la seva informació bàsica.

  - Funcionament:
    a) Es defineix, mitjançant un cursor, la informació que es vol obtenir.
    b) Indica el tipus de llistat (ROWTYPE).
    c) Verifica que s'introdueix un valor correcte en el nom del curs que es demana.
    d) Crea la capçalera del llistat.
    e) Mitjançant un bucle, obté cada una de les files resultants de la consulta definida en el cursor.
    f) Obté el nombre total de registres retornats.
    g) Si el procés és correcte, actualitza la taula `Log`, indicant en el paràmetre de sortida OK.
    h) Mostra el resultat de la consulta.
    i) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada
    `p_nomCurs IN Varchar2`: Nom del curs a què es refereix la informació.

  - Paràmetre de sortida
    RSP `OUT Varchar2`: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **`llistatAmonestacionsSancions`**: Procediment de consulta que permet obtenir un llistat de tots els tipus d'amonestacions o de sancions disponibles per a aplicar en el centre.

  - Funcionament:
    a) Declara un cursor per a obtenir dades de les amonestacions.
    b) Declara un altre cursor per a obtenir dades de les sancions.
    c) Indica el tipus de llistat (ROWTYPE).
    d) Verifica que s'introdueix un valor correcte per seleccionar el cursor: 'Amonestacions' o 'Sancions'.
    e) Executa el cursor d'acord amb el tipus de llistat seleccionat:
       i. Primer, crea la capçalera.

ii. Seguidament, mitjançant un bucle, obté cada una de les files resultants de la consulta definida en el cursor.
iii. Obté el nombre de registres retornats.

f) Si el procés és correcte, actualitza la taula Log, indicant en el paràmetre de sortida OK.
g) Mostra el resultat de la consulta.
h) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

- Paràmetres d'entrada

  p_amonestacio_sancio IN Varchar2: Indica la selecció de llistat que es vol obtenir: amonestacions o sancions.

- Paràmetre de sortida

  RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

- **llistatAmonSancionsAlumne**: Procediment de consulta que permet obtenir un llistat de totes les amonestacions i sancions d'un alumne.

  - Funcionament:

    a) Declara un cursor per a obtenir les amonestacions de l'alumne.
    b) Declara un altre cursor per a obtenir les sancions de l'alumne.
    c) Indica el tipus de llistat (ROWTYPE).
    d) Verifica que s'introdueix un valor correcte per seleccionar el cursor: 'Amonestacions' o 'Sancions'.
    e) Comprova que existeix l'alumne a què es fa referència.
    f) Executa el cursor segons el tipus de llistat seleccionat:
       i. Primer, crea la capçalera.
       ii. Seguidament, mitjançant un bucle, obté cada una de les files resultants de la consulta definida en el cursor.
       iii. Obté el nombre de registres retornats.
    g) Si el procés és correcte, actualitza la taula Log, indicant en el paràmetre de sortida OK.
    h) Mostra el resultat de la consulta.
    i) Si no es pot executar de forma adequada, llença l'excepció corresponent, indicant-ne en el paràmetre de sortida una descripció.

  - Paràmetres d'entrada

    p_amonestacio_sancio IN Varchar2: Indica la selecció de llistat que es vol obtenir: amonestacions o sancions.

p_alumne IN Number: Alumne de qui es vol obtenir el llistat.

- Paràmetre de sortida
  RSP OUT Varchar2: Resultat de l'execució: 'OK' si ha estat correcta o 'ERROR + descripció' si no ho ha estat

(índex)

8. PLA DE PROVES

Per comprovar el bon funcionament del sistema, s'ha desenvolupat un joc de proves mitjançant el qual verificar que els processos s'executen correctament i que es compleixen els requeriments establerts en el disseny. Així, s'inclouen aquells *scripts* que permeten introduir una sèrie de dades inicials per, posteriorment, comprovar, mitjançant l'execució dels procediments emmagatzemats, que es compleixen les funcionalitats i les restriccions marcades en el projecte. Aquestes mateixes dades permeten, també, comprovar el funcionament del mòdul estadístic, de manera que es pugui accedir a les dades en temps constant 1.

Les proves realitzades permeten testar totes les restriccions imposades al sistema, provocant els diferents casos d'error possibles per verificar-ne el comportament.

8.1. Càrrega de dades

A la carpeta del producte lliurat es pot trobar una subcarpeta anomenada "Inserts i proves", en la qual s'han inclòs diversos fitxers que permeten, d'una banda, fer la càrrega inicial de dades i, de l'altra, provar els procediments.

Pel que fa a la càrrega de dades, l'script "*Proves 1 – inserts inicials*" conté les dades necessàries per inserir dades a les taules `Provincia, Comarca, Municipi, CentreEscolar, Dates, DiaSetmana i HoraLectiva.`

Les dades relatives a les províncies, comarques i municipis s'han obtingut del *Municat,* pàgina oficial del Departament de Governació i Relacions Institucionals de la Generalitat de Catalunya, que conté informació de tots els ens locals de Catalunya.

Pel que fa a les dates, inicialment s'han inclòs en la taula els dies festius únicament del curs escolar 2010-2011. A més dels establerts oficialment, es consideren també com a festius els caps de setmana i els dies de vacances escolars. Quant a les festes locals, s'han pres com a exemple les de Barcelona. També s'hi han inclòs alguns registres de dates no festives per poder provar els procediments. Les altres dates que no són a la taula s'hi insereixen automàticament en donar d'alta al calendari escolar.

Per a carregar la resta de taules, cal executar els dos scripts següents:

- "*Proves 2 – test procediments alta 1*". Mitjançant aquest script s'aconsegueix proporcionar dades a les taules `Persona`, `Alumne`, `Professor`, `Curs`, `Assignatura`, `Amonestacio`, `Sancio` i `CalendariEscolar` i, alhora, testar el funcionament dels procediments d'alta.

- "*Proves 3 – test procediments alta 2*". Amb aquest script, en primer lloc s'insereixen dades de forma directa en les taules `Matriculat`, `Imparteix`, `Es_tutor` i `ReglaActivacio`, necessàries per poder provar els altres dos procediments d'alta, els quals, alhora, afegeixen les dades corresponents en les dues taules restants, `AmonestacionsEmeses` i `SancionsEmeses`.

(índex)

8.2. Execució del test

S'executen les proves prenent com a exemple un únic centre escolar. El resultat complet del test es mostra mitjançant la inclusió, a l'annex, del contingut de la taula *Log* i de diverses imatges de les taules estadístiques; no obstant això, es presenten algunes captures de pantalla per tal que serveixin com a exemples.

- En primer lloc, es comprova el funcionament correcte dels procediments d'alta mitjançant els dos scripts esmentats anteriorment; per tant, el resultat que s'espera d'aquesta execució és OK. Concretament, les operacions que s'efectuen són les següents:

  Proves 2 – test procediments alta 1
  - Es donen d'alta 10 persones.
  - D'aquestes, 7 es donen d'alta com a alumnes.
  - Les 3 restants es donen d'alta com a professors.
  - Es donen d'alta els 6 cursos possibles.
  - Es donen d'alta 3 exemples d'assignatures per cada un dels sis cursos.
  - Es donen d'alta 7 tipus d'amonestacions, incloent de tipus lleu, greu i molt greu.
  - Es donen d'alta 4 tipus de sancions, una com a personalitzada i les altre tres de les que s'activen automàticament.
  - Es donen d'alta 8 exemples de programacions al calendari escolar.

  Proves 3 – test procediments alta 2
  - Es donen d'alta 12 amonestacions a diversos alumnes de forma que puguin activar sancions automàtiques.
  - Es donen d'alta 2 sancions personalitzades.

  Tal com s'esperava i es pot comprovar a la taula *Log*, el resultat obtingut d'aquests tests és OK. Cal destacar, especialment, que mitjançant el primer *script* s'inicialitzen les diverses taules estadístiques i mitjançant el segon, s'omplen de contingut, tal com es pot comprovar en els exemples següents:

```
DECLARE
        RSP Varchar2(120) := '';
BEGIN

altaPersona ('Ramon','Clopés','Ricart','38444555V','Pg. de Sant Joan, 4', 948,93444555,'rcr@hotmail.com',RSP);
altaPersona ('Rosa','Clariana','Villar','38111111G','Cardener, 10', 948,93444995,'rcv@hotmail.com',RSP);
altaPersona ('Rafael','Millo','Rodríguez','40444886A','Ribes, 3', 948,93777995,'rmr@hotmail.com',RSP);
altaPersona ('Anna','Fresquet','Sánchez','38222222B','Rosselló, 320', 949,93444666,'afs@hotmail.com',RSP);
altaPersona ('Pere','Torné','Riu','38333333C','València, 546', 949,93222333,'ptr@hotmail.com',RSP);
altaPersona ('Marc','Martínez','Brugués','38444555D','Cantàbria, 192', 91,'93444546','mmb@hotmail.com',RSP);
altaPersona ('Lluís','Martínez','Pérez','38444001E','Pau Claris, 18', 948,93444001,'lmp@hotmail.com',RSP);
altaPersona ('Àlex','Costa','Costa','38555555F','Bailèn, 45', 948,93555555,'acc@hotmail.com',RSP);
altaPersona ('Neus','Ferrer','López','38666777G','Diputació, 420', 948,93666777,'nfl@hotmail.com',RSP);
altaPersona ('Clara','Bruguera','Casals','40888333H','Consell de Cent', 949,93888443,'cbc@hotmail.com',RSP);

altaAlumne (1,'38444A','20/08/1996','2008','Joan','Maria',RSP);
altaAlumne (2,'38555B','16/02/1996','2008','Pere','Joana',RSP);
altaAlumne (3,'40444C','01/04/1997','2009','Josep Antoni','Anna',RSP);
altaAlumne (4,'40446D','23/12/1997','2009','Àlex','Neus',RSP);
altaAlumne (5,'40450E','03/03/1997','2009','Sergi','Pilar',RSP);
altaAlumne (6,'41116F','03/06/1998','2010','Manel','Anna',RSP);
altaAlumne (7,'41132G','12/07/1998','2010','Fulgencio','Carme',RSP);

altaProfessor (8,'Magisteri',RSP);
altaProfessor (9,'Filologia Catalana',RSP);
altaProfessor (10,'Magisteri',RSP);
```

Inicio > SQL > Comandos SQL

Confirmación Automática Mostrar 30

```
select * from AmonestacionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	NOMBREAMONALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	0	-
2	Rosa Clariana Villar	0	-
3	Rafael Millo Rodríguez	0	-
4	Anna Fresquet Sánchez	0	-
5	Pere Torné Riu	0	-
6	Marc Martínez Brugués	0	-
7	Lluís Martínez Pérez	0	-

Enter SQL Statement:

```
DECLARE
        RSP Varchar2(120) := '';
BEGIN

altaAmonestacionsEmeses (1,8,1,1,3,'26/11/2010','10:28','N',RSP);
altaAmonestacionsEmeses (1,8,1,1,3,'14/12/2010','09:10','N',RSP);
altaAmonestacionsEmeses (1,8,4,1,3,'07/02/2011','12:32','N',RSP);
altaAmonestacionsEmeses (1,9,1,1,1,'21/02/2011','16:38','N',RSP);
altaAmonestacionsEmeses (2,9,2,1,1,'15/03/2011','10:07','N',RSP);
altaAmonestacionsEmeses (2,9,3,1,1,'16/03/2011','13:12','N',RSP);
altaAmonestacionsEmeses (5,9,5,2,1,'27/04/2011','08:30','N',RSP);
altaAmonestacionsEmeses (4,10,6,2,2,'29/04/2011','11:17','N',RSP);
altaAmonestacionsEmeses (4,9,7,2,1,'03/05/2011','12:05','N',RSP);
altaAmonestacionsEmeses (4,10,7,2,2,'06/05/2011','14:18','N',RSP);
altaAmonestacionsEmeses (5,8,6,2,3,'16/05/2011','09:16','N',RSP);
```

```
select * from AmonestacionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	NOMBREAMONALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	4	-
2	Rosa Clariana Villar	2	-
3	Rafael Millo Rodríguez	0	-
4	Anna Fresquet Sánchez	3	-
5	Pere Torné Riu	3	-
6	Marc Martínez Brugués	0	-
7	Lluís Martínez Pérez	0	-

- En segon lloc, es testeja el funcionament correcte dels procediments de consulta mitjançant l'execució, un a un, dels procediments inclosos en l'script "*Proves 4 - llistats*". El resultat esperat d'aquest test és la visualització dels llistats corresponents. S'adjunten en annex les captures de pantalla que verifiquen que l'execució ha estat correcta.

- Seguidament, es prova el funcionament correcte dels procediments de modificació i de baixa mitjançant l'execució de l'script "*Proves 5 - test procediments modif. i baixa*". Igualment, el resultat que s'espera d'aquest test es OK. El detall de les operacions que s'efectuen és el següent:

  Pel que fa als procediments de modificació:
  - Es modifica el primer cognom de la persona amb document identificatiu 38444001E. La modificació es reflecteix, també, en aquelles taules estadístiques en què hi consti.
  - Es modifica la data de naixement de l'alumne amb identificador núm. 4.
  - Es modifica la titulació del professor amb identificador núm. 9.
  - Es modifica el nom de l'assignatura amb identificadors 5, 3.
  - Es modifica el nombre de grups del curs 1r. Batxillerat.
  - Es modifica la gravetat del tipus d'amonestació amb identificador núm. 4
  - Es modifica el tipus de sanció amb identificador núm. 1.

- Es modifica l'assignatura i l'hora en què s'imparteix de la programació del calendari escolar identificada amb el número 7.

Pel que fa als procediments de baixa:

- Es dóna de baixa l'alumne amb document identificador 38444555V. Aquest alumne és un dels que té imposades amonestacions i sancions, de manera que s'actualitzen les estadístiques corresponents en el sentit d'afegir una observació indicativa de la baixa, però no s'elimina el registre de la taula.
- Es dóna de baixa el professor amb document identificador 40888333H. Igualment, aquest professor consta a diverses taules estadístiques, de manera que s'actualitzen en el sentit d'afegir l'observació que informa que s'ha donat de baixa, però no s'elimina la informació prèvia.
- Es dóna de baixa l'assignatura amb identificadors 6,3
- Es dóna de baixa el curs amb identificador núm. 6. La baixa d'aquest curs comporta la baixa de totes les assignatures que tenia assignades.
- Es dóna de baixa l'amonestació amb identificador núm. 5. No s'eliminen les referències estadístiques prèvies.
- Es dóna de baixa la sanció amb identificador núm. 4. Igualment, no s'elimina cap referència estadística pel que fa a aquesta sanció.
- Es dóna de baixa la programació del calendari escolar que correspon a l'identificador núm. 1.

Tal com s'esperava i es pot comprovar a la taula *Log*, el resultat obtingut d'aquests tests és OK.

- Per últim, mitjançant l'execució de l'script "*Proves 6 – errors procediments*", es comprova que els procediments disposen d'una definició correcta d'excepcions, provocant els errors que les disparen. En aquest cas, el resultat esperat és el text descriptiu de l'error en el paràmetre de sortida RSP, la qual cosa, com es pot comprovar mitjançant el llistat de la taula *Log*, es produeix correctament.

(índex)

9. VALORACIÓ ECONÒMICA

D'acord amb la planificació especificada, la realització del projecte s'ha dut a terme en aproximadament 3 mesos, concretament, en 98 dies, dels quals 70 han estat laborables i 28 corresponen als caps de setmana. Tot i que, inicialment, havia estimat una dedicació d'1,5 hores diària de dilluns a divendres i de 2 hores els dissabtes i els diumenges, la complexitat del projecte, sobretot pel que fa a la part de la implementació, ha comportat que, finalment, la dedicació hagi estat superior, havent sigut, aproximadament, d'unes 2 hores diàries en dies laborables i de 4 els festius. Així,

doncs, el temps total dedicat al desenvolupament del projecte s'estima que ha estat de 252 hores.

Per determinar el cost total del projecte, es té en compte la doble tasca que es realitza, és a dir, es comptabilitza tant pel rol de directora del projecte, com pel de programadora en Oracle, però tenint present la manca d'experiència en aquest àmbit laboral. D'acord amb diverses consultes a Internet sobre salaris, es pren com a base un salari mitjà de 40.000 euros bruts anuals. Atès que el sou anual brut correspon, normalment, a 14 mensualitats, i que cal aplicar-li les retencions corresponents a l'IRPF (suposem un 15%) i a la Seguretat Social (suposem un 4,80%), el sou net mensual resultant és de 2.312 €. Fent un càlcul amb 22 dies laborables al mes i amb 8 hores de treball diari, resulta un preu aproximat per hora de 13,14 €/h., que cal multiplicar per les 252 hores dedicades. (Web consultada: http://cuantocobro.blogspot.es/)

Així, doncs, la valoració econòmica resultant del projecte és, arrodonint, de 3.311 euros, la qual cosa suposa un increment aproximat del 56% sobre el pressupost que havia previst en un inici.

(índex)

10. CONCLUSIONS

Del resultat de les proves realitzades es pot extreure, com a principal conclusió, que la base de dades dissenyada i implementada compleix els requisits marcats i, en conseqüència, s'han complert els objectius plantejats a l'inici.

A nivell personal, el treball m'ha suposat una experiència molt enriquidora, tot i les dificultats, per mi, d'implementar un projecte en la seva totalitat, atesos els meus limitats coneixements del llenguatge de programació utilitzat, provinent exclusivament de les poques assignatures en què s'ha fet servit durant els estudis d'ETIG. No obstant això, el fet d'aprendre a desenvolupar projectes d'aquestes característiques va ser el principal objectiu que em va impulsar a iniciar aquests estudis, ja que la meva professió actual no té res a veure amb el món de la informàtica, tot i fer servir bases de dades de manera habitual.

Crec que aquesta inexperiència meva es veu reflectida, també, en l'exagerada desviació que ha sofert la valoració econòmica final del projecte respecte a la que havia previst inicialment, derivada bàsicament del còmput d'hores de dedicació, que ha estat considerable.

(índex)

11. GLOSSARI

Administrador de base de dades: Tipus d'usuari que realitza funcions centralitzades d'administració i control de la base de dades que asseguren que la seva explotació sigui correcta.

Atribut: Propietat d'una entitat.

Base de dades: Conjunt estructurat de dades que representa entitats i les seves interrelacions. La representació és única, integrada, però permet utilitzacions diverses i simultànies.

Cardinalitat: Manera de descriure el nombre d'entitats amb què està relacionada cada instància d'una relació.

Cas d'ús: Documentació d'una interacció entre el programari i el conjunt de papers d'una entitat exterior en relació amb el sistema considerat.

Clau primària: Atribut o conjunt d'atributs que permet distingir un objecte d'un altre.

Clau forana: Atribut que permet referenciar una clau primària i, en conseqüència, establir connexions entre les dades de les diferents relacions.

Diagrama E/R: Notació que permet representar gràficament les entitats, els seus atributs i les seves interrelacions.

Disseny conceptual: Etapa del disseny d'una base de dades que obté una estructura de la informació de la futura base de dades independent de la tecnologia que s'ha d'utilitzar.

Disseny lògic: Etapa del disseny d'una base de dades que parteix del resultat del disseny conceptual i el transforma per tal que s'adapti al model del sistema de gestió de base de dades amb què s'ha d'implementar la base de dades.

Disparador: Acció que s'executa automàticament quan es porta a terme un determinat esdeveniment d'inserció, esborrat o modificació sobre alguna taula de la base de dades.

Entitat: Conceptualització d'un objecte del món real que es pot distingir de la resta d'objectes i del quan interessen algunes propietats.

Entitat dèbil: Entitat que no és identificada completament pels seus atributs, sinó només de manera parcial.

Entitat subclasse: Especialització d'una classe general anomenada *superclasse* que permet modelitzar característiques específiques i pròpies.

Excepció: Estructura que serveix per controlar el tractament d'errors, permetent definir el comportament del programa.

Funció: En aquest context, agrupació d'instruccions que permet escriure-les una única vegada.

Interrelació: Associació entre entitats.

Log: Registre oficial d'esdeveniments durant un temps determinat.

Procediment emmagatzemat: Programa definit per l'usuari dins de la base de dades que executa una acció o conjunt d'accions especifiques. Té un nom, un conjunt de paràmetres (opcional) i un bloc de codi i pot retornar cap, un o més valors.

Regla de negoci: En el context d'una base de dades, qualsevol restricció, necessitat, requeriment o activitat especial que ha de ser verificada en el moment de gravar, esborrar o actualitzar informació.

Relació: Element d'una base de dades relacional que permet emmagatzemar dades relacionades entre sí.

Requisits: Descripció del comportament, propietats i restriccions del programari.

Seqüència: En aquest context, proporciona una llista consecutiva de nombres que serveix per simplificar les tasques de programació. La primera vegada que es crida la seqüència es retorna un valor determinat; en les consultes següents, s'obté un valor incrementat segons el tipus d'increment especificat.

Taula: Estructura de la base de dades que correspon a la relació del model relacional.

TFC: Treball final de carrera

(índex)

12. BIBLIOGRAFIA

- Documentació UOC:

- Mòduls didàctics associats al TFC (any acadèmic 2010-2011 - 2)
- Mòduls didàctics de l'assignatura Base de Dades I (any acadèmic 2006-2007 - 2)
- Mòduls didàctics de l'assignatura Base de Dades II (any acadèmic 2007-2008 - 1)
- Mòduls didàctics de l'assignatura Sistema de Gestió de Base de Dades (any acadèmic 2007-2008 – 2)
- Mòduls didàctics de l'assignatura Enginyeria de Programari (any acadèmic 2008-2009 – 1)
- Altres TFC d'anys anteriors facilitats com a exemple pel client.

- Pàgines web:

- http://www.postgresql.org/docs/9.0/interactive/errcodes-appendix.html
- http://www.java2s.com/Tutorial/Oracle/0500__Cursor/Catalog0500__Cursor.htm
- http://www.oracle.com/es/index.html
- http://www.municat.gencat.cat/
- http://cuantocobro.blogspot.es/

(índex)

13. ANNEXOS

Pel que fa a les proves realitzades, s'adjunten diverses captures de pantalla descriptives dels seu resultat, així com el contingut complet de la taula *Log*. A més, tot i que s'aporten els corresponents arxius `.sql`, s'adjunta en aquest annex l'especificació de les taules, seqüències, disparadors, funcions i procediments emmagatzemats, amb un determinat format per tal de facilitar-ne la revisió.

13.1. Imatges del resultat de les proves

1. Taula del nombre d'amonestacions dels alumnes, inicial i després d'imposar-ne

```
select * from AmonestacionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	NOMBREAMONALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	0	-
2	Rosa Clariana Villar	0	-
3	Rafael Millo Rodríguez	0	-
4	Anna Fresquet Sánchez	0	-
5	Pere Torné Riu	0	-
6	Marc Martínez Brugués	0	-
7	Lluís Martínez Pérez	0	-

```
select * from AmonestacionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	NOMBREAMONALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	4	-
2	Rosa Clariana Villar	2	-
3	Rafael Millo Rodríguez	0	-
4	Anna Fresquet Sánchez	3	-
5	Pere Torné Riu	3	-
6	Marc Martínez Brugués	0	-
7	Lluís Martínez Pérez	0	-

- Taula del nombre d'amonestacions dels alumnes en donar de baixa un d'ells

```
select * from AmonestacionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	NOMBREAMONALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	4	Alumne donat de baixa
2	Rosa Clariana Villar	2	-
3	Rafael Millo Rodríguez	0	-
4	Anna Fresquet Sánchez	3	-
5	Pere Torné Riu	3	-
6	Marc Martínez Brugués	0	-
7	Lluís Hernández Pérez	0	-

2. Taula de sancions per alumne i any, inicial i després d'imposar-ne

```
select * from SancionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	IDANY	NOMBRESANCIONSALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	2010	0	-
1	Ramon Clopés Ricart	2011	0	-
2	Rosa Clariana Villar	2010	0	-
2	Rosa Clariana Villar	2011	0	-
3	Rafael Millo Rodríguez	2010	0	-
3	Rafael Millo Rodríguez	2011	0	-
4	Anna Fresquet Sánchez	2010	0	-
4	Anna Fresquet Sánchez	2011	0	-
5	Pere Torné Riu	2010	0	-
5	Pere Torné Riu	2011	0	-
6	Marc Martínez Brugués	2010	0	-
6	Marc Martínez Brugués	2011	0	-
7	Lluís Martínez Pérez	2010	0	-
7	Lluís Martínez Pérez	2011	0	-

```
select * from SancionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	IDANY	NOMBRESANCIONSALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	2010	1	-
1	Ramon Clopés Ricart	2011	2	-
2	Rosa Clariana Villar	2010	0	-
2	Rosa Clariana Villar	2011	0	-
3	Rafael Millo Rodríguez	2010	0	-
3	Rafael Millo Rodríguez	2011	0	-
4	Anna Fresquet Sánchez	2010	0	-
4	Anna Fresquet Sánchez	2011	2	-
5	Pere Torné Riu	2010	0	-
5	Pere Torné Riu	2011	0	-
6	Marc Martínez Brugués	2010	0	-
6	Marc Martínez Brugués	2011	1	-
7	Lluís Martínez Pérez	2010	0	-
7	Lluís Martínez Pérez	2011	0	-

- Taula de sancions per alumne i any després de donar de baixa un alumne

```
select * from SancionsAlumne
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	IDANY	NOMBRESANCIONSALUMNE	OBSERVACIONS
1	Ramon Clopés Ricart	2010	1	Alumne donat de baixa
1	Ramon Clopés Ricart	2011	2	Alumne donat de baixa
2	Rosa Clariana Villar	2010	0	-
2	Rosa Clariana Villar	2011	0	-
3	Rafael Millo Rodríguez	2010	0	-
3	Rafael Millo Rodríguez	2011	0	-
4	Anna Fresquet Sánchez	2010	0	-
4	Anna Fresquet Sánchez	2011	2	-
5	Pere Torné Riu	2010	0	-
5	Pere Torné Riu	2011	0	-
6	Marc Martínez Brugués	2010	0	-
6	Marc Martínez Brugués	2011	1	-
7	Lluís Hernández Pérez	2010	0	-
7	Lluís Hernández Pérez	2011	0	-

3. Taula de la mitjana d'amonestacions per professor i any inicial i després d'imposar-ne

```
select * from AvgAmonestacionsProfAny
```

Resultados Explicar Describir SQL Guardado Historial

IDPROFESSOR	NOMPROFESSOR	AVGAMONPROFANY	OBSERVACIONS
8	Àlex Costa Costa	0	-
9	Neus Ferrer López	0	-
10	Clara Bruguera Casals	0	-

```
select * from AvgAmonestacionsProfAny
```

Resultados Explicar Describir SQL Guardado Historial

IDPROFESSOR	NOMPROFESSOR	AVGAMONPROFANY	OBSERVACIONS
8	Àlex Costa Costa	2,5	-
9	Neus Ferrer López	4	-
10	Clara Bruguera Casals	3	-

- Taula de la mitjana d'amonestacions per professor i any en donar un de baixa

```
select * from AvgAmonestacionsProfAny
```

Resultados Explicar Describir SQL Guardado Historial

IDPROFESSOR	NOMPROFESSOR	AVGAMONPROFANY	OBSERVACIONS
8	Àlex Costa Costa	2,5	-
9	Neus Ferrer López	4	-
10	Clara Bruguera Casals	3	Professor donat de baixa

4. Taula de sancions per curs i any, inicial i després d'imposar-ne

```
select * from SancionsCursAny
```

Resultados Explicar Describir SQL Guardado Historial

IDCURS	NOMCURS	IDANY	NOMBRESANCIONSCURS	OBSERVACIONS
1	1r. ESO	2010	0	-
1	1r. ESO	2011	0	-
2	2n. ESO	2010	0	-
2	2n. ESO	2011	0	-
3	3r. ESO	2010	0	-
3	3r. ESO	2011	0	-
4	4t. ESO	2010	0	-
4	4t. ESO	2011	0	-
5	1r. Batxillerat	2010	0	-
5	1r. Batxillerat	2011	0	-
6	2n. Batxillerat	2010	0	-
6	2n. Batxillerat	2011	0	-

```
select * from SancionsCursAny
```

Resultados Explicar Describir SQL Guardado Historial

IDCURS	NOMCURS	IDANY	NOMBRESANCIONSCURS	OBSERVACIONS
1	1r. ESO	2010	1	-
1	1r. ESO	2011	2	-
2	2n. ESO	2010	0	-
2	2n. ESO	2011	2	-
3	3r. ESO	2010	0	-
3	3r. ESO	2011	1	-
4	4t. ESO	2010	0	-
4	4t. ESO	2011	0	-
5	1r. Batxillerat	2010	0	-
5	1r. Batxillerat	2011	0	-
6	2n. Batxillerat	2010	0	-
6	2n. Batxillerat	2011	0	-

- Taula de sancions per curs i any després de donar de baixa un curs

```
select * from SancionsCursAny
```

Resultados Explicar Describir SQL Guardado Historial

IDCURS	NOMCURS	IDANY	NOMBRESANCIONSCURS	OBSERVACIONS
1	1r. ESO	2010	1	-
1	1r. ESO	2011	2	-
2	2n. ESO	2010	0	-
2	2n. ESO	2011	2	-
3	3r. ESO	2010	0	-
3	3r. ESO	2011	1	-
4	4t. ESO	2010	0	-
4	4t. ESO	2011	0	-
5	1r. Batxillerat	2010	0	-
5	1r. Batxillerat	2011	0	-
6	2n. Batxillerat	2010	0	Curs donat de baixa
6	2n. Batxillerat	2011	0	Curs donat de baixa

5. Taula de l'alumne més sancionat per anys

```
select * from AlumneMesSancionat;
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	IDANY	OBSERVACIONS
1	Ramon Clopés Ricart	2010	-
9999	No hi ha un únic alumne amb un màxim de sancions	2011	-

- Taula de l'alumne més sancionat després de donar de baixa un alumne

```
select * from AlumneMesSancionat
```

Resultados Explicar Describir SQL Guardado Historial

IDALUMNE	NOMALUMNE	IDANY	OBSERVACIONS
9999	No hi ha un únic alumne amb un màxim de sancions	2011	-
1	Ramon Clopés Ricart	2010	Alumne donat de baixa

6. Taula del professor més amonestador per curs

```
select * from ProfessorMesAmonestador
```

Resultados Explicar Describir SQL Guardado Historial

IDPROFESSOR	NOMPROFESSOR	IDCURS	NOMCURS	OBSERVACIONS
8	Àlex Costa Costa	1	1r. ESO	-
10	Clara Bruguera Casals	2	2n. ESO	-

- Taula del professor més amonestador per curs després de donar-ne un de baixa

```
select * from ProfessorMesAmonestador
```

Resultados Explicar Describir SQL Guardado Historial

IDPROFESSOR	NOMPROFESSOR	IDCURS	NOMCURS	OBSERVACIONS
8	Àlex Costa Costa	1	1r. ESO	-
10	Clara Bruguera Casals	2	2n. ESO	Professor donat de baixa

7. Taula de la mitjana d'amonestacions per curs, inicial i després d'imposar-ne

```
select * from AvgSancionsCurs
```

Resultados Explicar Describir SQL Guardado Historial

IDCURS	NOMCURS	AVGSANCIONSCURS	OBSERVACIONS
1	1r. ESO	2	-
2	2n. ESO	1,5	-
3	3r. ESO	1,6666666666666666666666666666666666666667	-
4	4t. ESO	0	-
5	1r. Batxillerat	0	-
6	2n. Batxillerat	0	-


```
select * from AvgSancionsCurs
```

Resultados Explicar Describir SQL Guardado Historial

IDCURS	NOMCURS	AVGSANCIONSCURS	OBSERVACIONS
1	1r. ESO	2	-
2	2n. ESO	1,5	-
3	3r. ESO	1,6666666666666666666666666666666666666667	-
4	4t. ESO	0	-
5	1r. Batxillerat	0	-
6	2n. Batxillerat	0	-

- Taula de la mitjana d'amonestacions per curs i any en donar un de baixa

```
select * from AvgSancionsCurs
```

Resultados Explicar Describir SQL Guardado Historial

IDCURS	NOMCURS	AVGSANCIONSCURS	OBSERVACIONS
1	1r. ESO	2	-
2	2n. ESO	1,5	-
3	3r. ESO	1,6666666666666666666666666666666666666667	-
4	4t. ESO	0	-
5	1r. Batxillerat	0	-
6	2n. Batxillerat	0	Curs donat de baixa

8. Taula inicial d'alumnes no amonestats

```
select * from AlumnesNoAmonestats
```

Resultados Explicar Describir SQL Guardac

NOMBREALNOAMON
7

- Taula final d'alumnes no amonestats

```
select * from AlumnesNoAmonestats;
```

Resultados Explicar Describir SQL Guardadc

NOMBREALNOAMON
3

9. Llistat d'alumnes d'un curs

```
DECLARE
        RSP Varchar2(120) := '';
BEGIN
        llistatAlumnesCurs('1r. ESO',RSP);
END;
```

Resultados Explicar Describir SQL Guardado Historial

```
Curs          Alumne                  Doc.identificació Núm.expedient Data de naixement Adreça                C.Postal Municipi  Comarca     Provincia  Telèfon   eMail                 Ingrés Nom del pare Nom de la mare
------------- ----------------------- ----------------- ------------- ---------------- --------------------- -------- ---------- ----------- ---------- --------- --------------------- ------ ------------ -------------
1r. ESO       Rosa Clariana Villar    3811111116Q       38555B        16/02/96         Cardener, 10          08009    Barcelona  Barcelonès  Barcelona  93444995  rcv@hotmail.com       2008   Pere         Joana
1r. ESO       Ramon Clopés Ricart     38444555V         38444A        20/08/96         Pg. de Sant Joan, 4   08009    Barcelona  Barcelonès  Barcelona  93444555  rcr@hotmail.com       2008   Joan         Maria
1r. ESO       Rafael Millo Rodriguez  40444886A         40444C        01/04/97         Ribes, 3              08009    Barcelona  Barcelonès  Barcelona  93777995  rmr@hotmail.com       2009   Josep Antoni Anna

Total d'alumnes del curs: 3
```

10. Llistat d'amonestacions imposades

```
DECLARE
        RSP Varchar2(120) := '';
BEGIN
       llistatAmonestacionsImposades(RSP);
END;
```

Resultados Explicar Describir SQL Guardado Historial

```
Alumne                    Tipus d'amonestació       Gravetat   Data      Hora   Curs          Assignatura         Professor                 Comunicat pares
------------------------- ------------------------- ---------- --------- ------ ------------- ------------------- ------------------------- ---------------
Pere Torné Riu            Alumne_no_respectuós      Greu       23/11/10  09:16  2n. ESO       Matemàtiques        Àlex Costa Costa                N
Ramon Clopés Ricart       Arriba_tard               Lleu       26/11/10  10:28  1r. ESO       Matemàtiques        Àlex Costa Costa                N
Ramon Clopés Ricart       Arriba_tard               Lleu       14/12/10  09:10  1r. ESO       Matemàtiques        Àlex Costa Costa                N
Ramon Clopés Ricart       Alumne_no_deures          Greu       07/02/11  12:32  1r. ESO       Matemàtiques        Àlex Costa Costa                N
Ramon Clopés Ricart       Arriba_tard               Lleu       21/02/11  16:38  1r. ESO       Llengua catalana    Neus Ferrer López               N
Rosa Clariana Villar      Alumne_soroll             Lleu       15/03/11  10:07  2n. ESO       Llengua castellana  Clara Bruguera Casals           N
Rosa Clariana Villar      Alumne_malparlat          Lleu       16/03/11  13:12  1r. ESO       Llengua catalana    Neus Ferrer López               N
Pere Torné Riu            Alumne_no_assistència     Greu       27/04/11  08:30  2n. ESO       Llengua catalana    Neus Ferrer López               N
Anna Fresquet Sánchez     Alumne_no_respectuós      Greu       29/04/11  11:17  2n. ESO       Llengua castellana  Clara Bruguera Casals           N
Anna Fresquet Sánchez     Alumne_agressiu_o_violent Molt greu  03/05/11  12:05  2n. ESO       Llengua catalana    Neus Ferrer López               N
Anna Fresquet Sánchez     Alumne_agressiu_o_violent Molt greu  06/05/11  14:18  2n. ESO       Llengua castellana  Clara Bruguera Casals           N
Pere Torné Riu            Alumne_no_respectuós      Greu       16/05/11  09:16  2n. ESO       Matemàtiques        Àlex Costa Costa                N

Total d'amonestacions imposades = 12
```

11. Llistat d'amonestacions i de sancions disponibles

```
BEGIN
        llistatAmonestacionsSancions('Amonestacions',RSP);
        llistatAmonestacionsSancions('Sancions',RSP);
END;
```

Resultados Explicar Describir SQL Guardado Historial

```
Tipus d'amonestació                                     Descripció de l'amonestacio                                                  Gravetat
------------------------------------------------------- ---------------------------------------------------------------------------- ----------
Arriba_tard                                             L'alumne arriba tard a classe                                                Lleu
Alumne_soroll                                           L'alumne fa molt de soroll en classe                                         Lleu
Alumne_malparlat                                        L'alumne és malparlat                                                        Lleu
Alumne_no_deures                                        L'alumne no fa els deures                                                    Greu
Alumne_no_assistència                                   L'alumne roman fora de l'aula sense autorització                             Greu
Alumne_no_respectuós                                    L'alumne es mostra irrespectuós amb el professor o els companys              Greu
Alumne_agressiu_o_violent                               L'alumne protagonitza algun acte de violència                                Molt greu

Total tipus d'amonestacions: 7

Tipus de sanció                                         Motiu d'imposició
------------------------------------------------------- ------------------------------------------------------------------------------
Quedar-se una hora extra d'estudi durant una setmana    Automàticament quan l'alumne arriba tard tres vegades
Prova d'avaluació extra                                 Automàticament quan fan molt de soroll en classe 5 vegades
Expulsió 2 dies                                         Automàticament en tenir una amonestació de tipus molt greu
Personalitzada: repetir treball                         L'alumne no ha treballat prou la feina requerida

Total tipus de sancions: 4

Sentencia procesada.
```

12. Llistat d'amonestacions i de sancions d'un alumne

```
DECLARE
        RSP Varchar2(120) := '';
BEGIN
       llistatAmonSancionsAlumne('Amonestacions',1,RSP);
       llistatAmonSancionsAlumne('Sancions',1,RSP);
END;
```

Resultados Explicar Describir SQL Guardado Historial

```
Alumne                          Tipus d'amonestació       Motiu de l'amonestació                      Gravetat       Data
------------------------------- ------------------------- ------------------------------------------- -------------- ----------
Ramon Clopés Ricart             Arriba_tard               L'alumne arriba tard a classe               Lleu           26/11/10
Ramon Clopés Ricart             Arriba_tard               L'alumne arriba tard a classe               Lleu           14/12/10
Ramon Clopés Ricart             Alumne_no_deures          L'alumne no fa els deures                   Greu           07/02/11
Ramon Clopés Ricart             Arriba_tard               L'alumne arriba tard a classe               Lleu           21/02/11

Total d'amonestacions de l'alumne : 4
Alumne                          Tipus de sanció                                    Motiu de la sanció                                                                  Data
------------------------------- -------------------------------------------------- ----------------------------------------------------------------------------------- ----------
Ramon Clopés Ricart             Personalitzada: repetir treball                    L'alumne no ha treballat prou la feina requerida                                    29/11/10
Ramon Clopés Ricart             Quedar-se una hora extra d'estudi durant una setmana Automàticament quan l'alumne arriba tard tres vegades                            21/02/11
Ramon Clopés Ricart             Personalitzada: repetir treball                    L'alumne no ha treballat prou la feina requerida                                    28/04/11

Total de sancions de l'alumne: 3
```

13. Taula *Log*

IDLOG	DATA	PROCEDIMENT	ENTRADA	SORTIDA
1	04-JUN-11 09.47.30,000000 PM	altaPersona	Ramon, Clopés, Ricart, 38444555V, Pg. de Sant Joan, 4, 948, 93444555, rcr@hotmail.com	OK
2	04-JUN-11 09.47.30,000000 PM	altaPersona	Rosa, Clariana, Villar, 381111116Q, Cardener, 10, 948, 93444995, rcv@hotmail.com	OK
3	04-JUN-11 09.47.30,000000 PM	altaPersona	Rafael, Millo, Rodríguez, 40444886A, Ribes, 3, 948, 93777995, rmr@hotmail.com	OK
4	04-JUN-11 09.47.30,000000 PM	altaPersona	Anna, Fresquet, Sánchez, 38222222B, Rosselló, 320, 949, 93444666, afs@hotmail.com	OK
5	04-JUN-11 09.47.30,000000 PM	altaPersona	Pere, Torné, Riu, 38333333C, València, 546, 949, 93222333, ptr@hotmail.com	OK
6	04-JUN-11 09.47.30,000000 PM	altaPersona	Marc, Martínez, Brugués, 38444555D, Cantàbria, 192, 91, 93444546, mmb@hotmail.com	OK
7	04-JUN-11 09.47.30,000000 PM	altaPersona	Lluís, Martínez, Pérez, 38444001E, Pau Claris, 18, 948, 93444001, lmp@hotmail.com	OK
8	04-JUN-11 09.47.30,000000 PM	altaPersona	Àlex, Costa, Costa, 38555555F, Bailèn, 45, 948, 93555555, acc@hotmail.com	OK
9	04-JUN-11 09.47.30,000000 PM	altaPersona	Neus, Ferrer, López, 38666777G, Diputació, 420, 948, 93666777, nfl@hotmail.com	OK
10	04-JUN-11 09.47.30,000000 PM	altaPersona	Clara, Bruguera, Casals, 40888333H, Consell de Cent, 949, 93888443, cbc@hotmail.com	OK
11	04-JUN-11 09.47.30,000000 PM	altaAlumne	1, 38444A, 20/08/96, 2008, Joan, Maria	OK
12	04-JUN-11 09.47.30,000000 PM	altaAlumne	2, 38555B, 16/02/96, 2008, Pere, Joana	OK

13	04-JUN-11 09.47.30,000000 PM	altaAlumne	3, 40444C, 01/04/97, 2009, Josep Antoni, Anna	OK
14	04-JUN-11 09.47.30,000000 PM	altaAlumne	4, 40446D, 23/12/97, 2009, Àlex, Neus	OK
15	04-JUN-11 09.47.30,000000 PM	altaAlumne	5, 40450E, 03/03/97, 2009, Sergi, Pilar	OK
16	04-JUN-11 09.47.30,000000 PM	altaAlumne	6, 41116F, 03/06/98, 2010, Manel, Anna	OK
17	04-JUN-11 09.47.30,000000 PM	altaAlumne	7, 41132G, 12/07/98, 2010, Fulgencio, Carme	OK
18	04-JUN-11 09.47.30,000000 PM	altaProfessor	8, Magisteri	OK
19	04-JUN-11 09.47.30,000000 PM	altaProfessor	9, Filologia Catalana	OK
20	04-JUN-11 09.47.30,000000 PM	altaProfessor	10, Magisteri	OK
21	04-JUN-11 09.47.30,000000 PM	altaCurs	1r. ESO, 3	OK
22	04-JUN-11 09.47.30,000000 PM	altaCurs	2n. ESO, 3	OK
23	04-JUN-11 09.47.30,000000 PM	altaCurs	3r. ESO, 2	OK
24	04-JUN-11 09.47.30,000000 PM	altaCurs	4t. ESO, 2	OK
25	04-JUN-11 09.47.30,000000 PM	altaCurs	1r. Batxillerat, 2	OK
26	04-JUN-11 09.47.30,000000 PM	altaCurs	2n. Batxillerat, 2	OK
27	04-JUN-11 09.47.30,000000 PM	altaAssignatura	1, 1, Llengua catalana	OK
28	04-JUN-11 09.47.30,000000 PM	altaAssignatura	1, 2, Llengua castellana	OK
29	04-JUN-11 09.47.30,000000 PM	altaAssignatura	1, 3, Matemàtiques	OK
30	04-JUN-11 09.47.30,000000 PM	altaAssignatura	2, 1, Llengua catalana	OK
31	04-JUN-11 09.47.30,000000 PM	altaAssignatura	2, 2, Llengua castellana	OK
32	04-JUN-11 09.47.30,000000 PM	altaAssignatura	2, 3, Matemàtiques	OK
33	04-JUN-11 09.47.30,000000 PM	altaAssignatura	3, 1, Llengua catalana	OK
34	04-JUN-11 09.47.30,000000 PM	altaAssignatura	3, 2, Llengua castellana	OK
35	04-JUN-11 09.47.30,000000 PM	altaAssignatura	3, 3, Matemàtiques	OK
36	04-JUN-11 09.47.30,000000 PM	altaAssignatura	4, 1, Llengua catalana	OK
37	04-JUN-11 09.47.30,000000 PM	altaAssignatura	4, 2, Llengua castellana	OK
38	04-JUN-11 09.47.30,000000 PM	altaAssignatura	4, 3, Matemàtiques	OK
39	04-JUN-11 09.47.30,000000 PM	altaAssignatura	5, 1, Llengua catalana	OK
40	04-JUN-11 09.47.30,000000 PM	altaAssignatura	5, 2, Llengua castellana	OK
41	04-JUN-11 09.47.30,000000 PM	altaAssignatura	5, 3, Matemàtiques	OK
42	04-JUN-11 09.47.30,000000 PM	altaAssignatura	6, 1, Llengua catalana	OK
43	04-JUN-11 09.47.30,000000 PM	altaAssignatura	6, 2, Llengua castellana	OK
44	04-JUN-11 09.47.30,000000 PM	altaAssignatura	6, 3, Matemàtiques	OK
45	04-JUN-11 09.47.30,000000 PM	altaAmonestacio	Arriba_tard, L'alumne arriba tard a classe, Lleu	OK
46	04-JUN-11 09.47.30,000000 PM	altaAmonestacio	Alumne_soroll, L'alumne fa molt de soroll en classe, Lleu	OK
47	04-JUN-11 09.47.30,000000 PM	altaAmonestacio	Alumne_malparlat, L'alumne és malparlat, Lleu	OK

48	04-JUN-11 09.47.30,000000 PM	altaAmonestacio	Alumne_no_deures, L'alumne no fa els deures, Greu	OK
49	04-JUN-11 09.47.30,000000 PM	altaAmonestacio	Alumne_no_assistència, L'alumne roman fora de l'aula sense autorització , Greu	OK
50	04-JUN-11 09.47.30,000000 PM	altaAmonestacio	Alumne_no_respectuós, L'alumne es mostra irrespectuós amb el professor o els companys, Greu	OK
51	04-JUN-11 09.47.30,000000 PM	altaAmonestacio	Alumne_agressiu_o_violent, L'alumne protagonitza algun acte de violència, Molt greu	OK
52	04-JUN-11 09.47.30,000000 PM	altaSancio	Quedar-se una hora extra d'estudi durant una setmana, Automàticament quan l'alumne arriba tard tres vegades	OK
53	04-JUN-11 09.47.30,000000 PM	altaSancio	Prova d'avaluació extra, Automàticament quan fan molt de soroll en classe 5 vegades	OK
54	04-JUN-11 09.47.30,000000 PM	altaSancio	Expulsió 2 dies, Automàticament en tenir una amonestació de tipus molt greu	OK
55	04-JUN-11 09.47.30,000000 PM	altaSancio	Personalitzada: repetir treball, L'alumne no ha treballat prou la feina requerida	OK
56	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 07/09/10, 1, 1, 1, 10	OK
57	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 07/09/10, 1, 2, 1, 9	OK
58	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 07/09/10, 2, 1, 1, 16	OK
59	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 08/09/10, 2, 2, 1, 11	OK
60	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 08/09/10, 3, 3, 2, 10	OK
61	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 10/01/11, 4, 2, 3, 12	OK
62	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 10/01/11, 5, 3, 2, 8	OK
63	04-JUN-11 09.47.30,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 10/01/11, 6, 1, 1, 12	OK
64	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	1	OK
65	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
66	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	8, 26/11/10	OK
67	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	8, 1	OK
68	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	1, 8, 1, 1, 3, 26/11/10, 10:28, N	OK
69	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	5	OK
70	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
71	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	8, 23/11/10	OK
72	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	8, 2	OK
73	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	5, 8, 6, 2, 3, 23/11/10, 09:16, N	OK
74	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	1	OK
75	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
76	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	8, 14/12/10	OK
77	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	8, 1	OK
78	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	1, 8, 1, 1, 3, 14/12/10, 09:10, N	OK
79	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	1	OK

80	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
81	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	8, 07/02/11	OK
82	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	8, 1	OK
83	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	1, 8, 4, 1, 3, 07/02/11, 12:32, N	OK
84	04-JUN-11 09.47.55,000000 PM	estSancionsAlumne	1, 21/02/11	OK
85	04-JUN-11 09.47.55,000000 PM	estAlumneMesSancionat	1, 21/02/11	OK
86	04-JUN-11 09.47.55,000000 PM	estAvgSancionsCurs	1	OK
87	04-JUN-11 09.47.55,000000 PM	estSancionsCursAny	1, 21/02/11	OK
88	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	1	OK
89	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
90	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	9, 21/02/11	OK
91	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	9, 1	OK
92	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	1, 9, 1, 1, 1, 21/02/11, 16:38, N	OK
93	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	2	OK
94	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
95	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	10, 15/03/11	OK
96	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	10, 2	No hi ha un únic professor com a màxim amonestador en aquest curs
97	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	2, 10, 2, 2, 2, 15/03/11, 10:07, N	OK
98	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	2	OK
99	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
100	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	9, 16/03/11	OK
101	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	9, 1	OK
102	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	2, 9, 3, 1, 1, 16/03/11, 13:12, N	OK
103	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	5	OK
104	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
105	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	9, 27/04/11	OK
106	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	9, 2	No hi ha un únic professor com a màxim amonestador en aquest curs
107	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	5, 9, 5, 2, 1, 27/04/11, 08:30, N	OK
108	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	4	OK
109	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
110	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	10, 29/04/11	OK
111	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	10, 2	OK
112	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	4, 10, 6, 2, 2, 29/04/11, 11:17, N	OK
113	04-JUN-11 09.47.55,000000 PM	estSancionsAlumne	4, 03/05/11	OK

114	04-JUN-11 09.47.55,000000 PM	estAlumneMesSancionat	4, 03/05/11	No hi ha un únic alumne amb un màxim de sancions per a aquest any
115	04-JUN-11 09.47.55,000000 PM	estAvgSancionsCurs	4	OK
116	04-JUN-11 09.47.55,000000 PM	estSancionsCursAny	4, 03/05/11	OK
117	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	4	OK
118	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
119	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	9, 03/05/11	OK
120	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	9, 2	No hi ha un únic professor com a màxim amonestador en aquest curs
121	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	4, 9, 7, 2, 1, 03/05/11, 12:05, N	OK
122	04-JUN-11 09.47.55,000000 PM	estSancionsAlumne	4, 06/05/11	OK
123	04-JUN-11 09.47.55,000000 PM	estAlumneMesSancionat	4, 06/05/11	OK
124	04-JUN-11 09.47.55,000000 PM	estAvgSancionsCurs	4	OK
125	04-JUN-11 09.47.55,000000 PM	estSancionsCursAny	4, 06/05/11	OK
126	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	4	OK
127	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
128	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	10, 06/05/11	OK
129	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	10, 2	OK
130	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	4, 10, 7, 2, 2, 06/05/11, 14:18, N	OK
131	04-JUN-11 09.47.55,000000 PM	estAmonestacionsAlumne	5	OK
132	04-JUN-11 09.47.55,000000 PM	estAlumnesNoAmon		OK
133	04-JUN-11 09.47.55,000000 PM	estAvgAmonProfAny	8, 16/05/11	OK
134	04-JUN-11 09.47.55,000000 PM	estProfMesAmon	8, 2	OK
135	04-JUN-11 09.47.55,000000 PM	altaAmonestacionsEmeses	5, 8, 6, 2, 3, 16/05/11, 09:16, N	OK
136	04-JUN-11 09.47.55,000000 PM	estSancionsAlumne	1, 29/11/10	OK
137	04-JUN-11 09.47.55,000000 PM	estSancionsCursAny	1, 29/11/10	OK
138	04-JUN-11 09.47.55,000000 PM	estAvgSancionsCurs	1	OK
139	04-JUN-11 09.47.55,000000 PM	estAlumneMesSancionat	1, 29/11/10	OK
140	04-JUN-11 09.47.55,000000 PM	altaSancionsEmeses	1, 8, 4, 29/11/10	OK
141	04-JUN-11 09.47.55,000000 PM	estSancionsAlumne	6, 28/04/11	OK
142	04-JUN-11 09.47.55,000000 PM	estSancionsCursAny	6, 28/04/11	OK
143	04-JUN-11 09.47.55,000000 PM	estAvgSancionsCurs	6	OK
144	04-JUN-11 09.47.55,000000 PM	estAlumneMesSancionat	6, 28/04/11	OK
145	04-JUN-11 09.47.55,000000 PM	altaSancionsEmeses	6, 8, 4, 28/04/11	OK
146	04-JUN-11 09.47.55,000000 PM	estSancionsAlumne	1, 28/04/11	OK
147	04-JUN-11 09.47.55,000000 PM	estSancionsCursAny	1, 28/04/11	OK

148	04-JUN-11 09.47.55,000000 PM	estAvgSancionsCurs	1	OK
149	04-JUN-11 09.47.55,000000 PM	estAlumneMesSancionat	1, 28/04/11	No hi ha un únic alumne amb un màxim de sancions per a aquest any
150	04-JUN-11 09.47.55,000000 PM	altaSancionsEmeses	1, 8, 4, 28/04/11	OK
151	04-JUN-11 09.49.25,000000 PM	llistatAmonestacionsImposades		OK
152	04-JUN-11 09.51.58,000000 PM	llistatAlumnesCurs	1r. ESO	OK
153	04-JUN-11 09.54.42,000000 PM	llistatAmonestacionsSancions	Amonestacions	OK
154	04-JUN-11 09.54.42,000000 PM	llistatAmonestacionsSancions	Sancions	OK
155	04-JUN-11 09.56.46,000000 PM	llistatAmonSancionsAlumne	Amonestacions, 1	OK
156	04-JUN-11 09.56.46,000000 PM	llistatAmonSancionsAlumne	Sancions, 1	OK
157	04-JUN-11 09.59.22,000000 PM	llistatAmonestacionsImposades		OK
158	04-JUN-11 10.00.30,000000 PM	modifPersona	Lluís, Hernández, Pérez, 38444001E, Pau Claris, 18, 948, 93444001, lmp@hotmail.com	OK
159	04-JUN-11 10.00.30,000000 PM	modifAlumne	4, 23/11/97, 2009, Àlex, Neus	OK
160	04-JUN-11 10.00.30,000000 PM	modifProfessor	9, Filologia Castellana	OK
161	04-JUN-11 10.00.30,000000 PM	modifAssignatura	5, 3, Història	OK
162	04-JUN-11 10.00.30,000000 PM	modifCurs	5, 1r. Batxillerat, 3	OK
163	04-JUN-11 10.00.30,000000 PM	modifAmonestacio	4, Alumne_no_deures, L'alumne no fa els deures, Lleu	OK
164	04-JUN-11 10.00.30,000000 PM	modifSancio	1, Quedar-se dues hores extres d'estudi durant una setmana, Automàticament quan l'alumne arriba tard tres vegades	OK
165	04-JUN-11 10.00.30,000000 PM	modifCalendariEscolar	7, calendari escolar 2010-2011, 10/01/11, 5, 1, 2, 13	OK
166	04-JUN-11 10.00.30,000000 PM	baixaPersona	38444555V	OK
167	04-JUN-11 10.00.30,000000 PM	baixaPersona	40888333H	OK
168	04-JUN-11 10.00.30,000000 PM	baixaAssignatura	6, 3	OK
169	04-JUN-11 10.00.30,000000 PM	baixaCurs	6	OK
170	04-JUN-11 10.00.30,000000 PM	baixaAmonestacio	5	OK
171	04-JUN-11 10.00.30,000000 PM	baixaSancio	2	OK
172	04-JUN-11 10.00.30,000000 PM	baixaCalendariEscolar	1	OK
173	04-JUN-11 10.02.09,000000 PM	altaPersona	Ramon, , Niubó, 39111222S, Casp, 19, 948, 93111222, rrn@hotmail.com	ERROR: Hi ha camps que no poden ser nuls
174	04-JUN-11 10.02.09,000000 PM	altaPersona	Lluís, Miró, García, 38444555D, Londres, 7, 949, 93999888, lmg@hotmail.com	ERROR: Aquesta persona ja existeix
175	04-JUN-11 10.02.09,000000 PM	altaPersona	Iván, Cerqueda, Vilana, 38443556V, Travessera de les Corts, 24, 960, 93443556, icv@hotmail.com	ERROR: El municipi indicat no existeix
176	04-JUN-11 10.02.09,000000 PM	altaAlumne	11, , 12/04/96, 2008, Ignasi, Elisabet	ERROR: Hi ha camps que no poden ser nuls
177	04-JUN-11 10.02.09,000000 PM	altaAlumne	12, 38444C, 15/09/96, 2008, Joan, Conxita	ERROR: Abans cal donar d'alta el registre a la taula Persona
178	04-JUN-11 10.02.09,000000 PM	altaAlumne	15, 39456T, 09/01/96, 2008, Ferran, Eva	ERROR: Abans cal donar d'alta el registre a la taula Persona

179	04-JUN-11 10.02.09,000000 PM	altaAlumne	9, 39547U, 01/07/96, 2008, Víctor, Victòria	ERROR: Aquesta persona ha estat prèviament donada d'alta com a professor/a
180	04-JUN-11 10.02.09,000000 PM	altaAlumne	2, 38444B, 20/08/96, 2008, Joan, Maria	ERROR: El número identificador de la persona està duplicat
181	04-JUN-11 10.02.09,000000 PM	altaProfessor	13	ERROR: Hi ha camps que no poden ser nuls
182	04-JUN-11 10.02.09,000000 PM	altaProfessor	3, Magisteri	ERROR: Aquesta persona ha estat prèviament donada d'alta com a alumne
183	04-JUN-11 10.02.09,000000 PM	altaProfessor	9, Magisteri	ERROR: El número identificador de la persona està duplicat
184	04-JUN-11 10.02.09,000000 PM	altaCurs	, 4	ERROR: Hi ha camps que no poden ser nuls
185	04-JUN-11 10.02.09,000000 PM	altaCurs	1r. ESO, 4	ERROR: Aquest curs ja existeix
186	04-JUN-11 10.02.09,000000 PM	altaCurs	5è. ESO, 3	ERROR: El nom del curs no és correcte
187	04-JUN-11 10.02.09,000000 PM	altaAssignatura	1, 4,	ERROR: Hi ha camps que no poden ser nuls
188	04-JUN-11 10.02.09,000000 PM	altaAssignatura	7, 1, Llengua catalana	ERROR: El curs indicat no existeix
189	04-JUN-11 10.02.09,000000 PM	altaAssignatura	1, 1, Llengua catalana	ERROR: Aquesta assignatura està duplicada
190	04-JUN-11 10.02.09,000000 PM	altaAmonestacio	, , Lleu	ERROR: Hi ha camps que no poden ser nuls
191	04-JUN-11 10.02.09,000000 PM	altaAmonestacio	Alumne_brut, L'alumne arriba a classe amb aspecte poc polit, Mitjà	ERROR: La gravetat de l'amonestació només pot ser Lleu, Greu o Molt greu
192	04-JUN-11 10.02.09,000000 PM	altaAmonestacio	Arriba_tard, L'alumne no és puntual, Lleu	ERROR: El tipus d'amonestació ja existeix
193	04-JUN-11 10.02.09,000000 PM	altaSancio	, L'alumne no ha treballat prou la feina requerida	ERROR: Hi ha camps que no poden ser nuls
194	04-JUN-11 10.02.09,000000 PM	altaSancio	Personalitzada: repetir treball, L'alumne no ha presentat els deures en el temps marcat	ERROR: El tipus de sanció ja existeix
195	04-JUN-11 10.02.09,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, , 1, 1, 1, 10	ERROR: Hi ha camps que no poden ser nuls
196	04-JUN-11 10.02.09,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 07/03/11, 1, 1, 1, 10	ERROR: La data indicada correspon a un dia festiu
197	04-JUN-11 10.02.09,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 07/09/10, 1, 6, 1, 16	ERROR: L'identificador del curs i/o de l'assignatura introduït no existeix
198	04-JUN-11 10.02.09,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 07/09/10, 1, 1, 1, 20	ERROR: L'hora i/o dia indicat no existeix
199	04-JUN-11 10.02.09,000000 PM	altaCalendariEscolar	calendari escolar 2010-2011, 07/09/10, 1, 2, 1, 9	ERROR: El registre de Calendari Escolar indicat ja existeix
200	04-JUN-11 10.02.09,000000 PM	altaAmonestacionsEmeses	1, , 1, 1, 3, 26/11/10, 10:28, N	ERROR: Hi ha camps que no poden ser nuls
201	04-JUN-11 10.02.09,000000 PM	altaAmonestacionsEmeses	20, 8, 1, 1, 3, 14/12/10, 09:10, N	ERROR: L'identificador de l'alumne indicat no existeix
202	04-JUN-11 10.02.09,000000 PM	altaAmonestacionsEmeses	3, 19, 1, 1, 3, 14/12/10, 09:10, N	ERROR: L'identificador del professor indicat no existeix
203	04-JUN-11 10.02.09,000000 PM	altaAmonestacionsEmeses	3, 9, 18, 1, 3, 14/12/10, 09:10, N	ERROR: L'identificador indicat de l'amonestació no existeix
204	04-JUN-11 10.02.09,000000 PM	altaAmonestacionsEmeses	3, 8, 1, 4, 2, 26/11/10, 10:28, N	ERROR: El professor indicat no imparteix l'assignatura assenyalada
205	04-JUN-11 10.02.09,000000 PM	altaAmonestacionsEmeses	3, 8, 1, 2, 3, 07/03/11, 10:28, N	ERROR: La data indicada correspon a un dia festiu
206	04-JUN-11 10.02.09,000000 PM	altaAmonestacionsEmeses	5, 8, 6, 2, 3, 23/11/10, 09:16, N	ERROR: El registre indicat d'AmonestacionsEmeses ja existeix
207	04-JUN-11 10.02.09,000000 PM	altaSancionsEmeses	6, , 4, 28/04/11	ERROR: Hi ha camps que no poden ser nuls

208	04-JUN-11 10.02.09,000000 PM	altaSancionsEmeses	20, 8, 4, 28/04/11	ERROR: L'identificador de l'alumne indicat no existeix
209	04-JUN-11 10.02.09,000000 PM	altaSancionsEmeses	3, 19, 4, 28/04/11	ERROR: L'identificador del professor indicat no existeix
210	04-JUN-11 10.02.09,000000 PM	altaSancionsEmeses	3, 8, 10, 28/04/11	ERROR: L'identificador indicat de la sanció no existeix
211	04-JUN-11 10.02.09,000000 PM	altaSancionsEmeses	3, 8, 4, 07/03/11	ERROR: La data indicada correspon a un dia festiu
212	04-JUN-11 10.02.09,000000 PM	altaSancionsEmeses	6, 8, 4, 28/04/11	ERROR: El registre indicat ja existeix a SancionsEmeses
213	04-JUN-11 10.02.09,000000 PM	baixaPersona	58444555V	ERROR: No s'ha trobat cap persona amb el document identificatiu indicat
214	04-JUN-11 10.02.09,000000 PM	baixaAssignatura	6	ERROR: Els identificadors de l'assignatura no poden ser nuls
215	04-JUN-11 10.02.09,000000 PM	baixaAssignatura	6, 9	ERROR: No s'ha trobat cap assignatura amb els identificadors indicats
216	04-JUN-11 10.02.09,000000 PM	baixaCurs	15	ERROR: No s'ha trobat cap curs amb l'identificador indicat
217	04-JUN-11 10.02.09,000000 PM	baixaAmonestacio	15	ERROR: No s'ha trobat cap amonestació amb l'identificador indicat
218	04-JUN-11 10.02.09,000000 PM	baixaSancio	20	ERROR: No s'ha trobat cap sanció amb l'identificador indicat
219	04-JUN-11 10.02.09,000000 PM	baixaCalendariEscolar	60	ERROR: No s'ha trobat cap registre en CalendariEscolar amb l'identificador indicat
220	04-JUN-11 10.02.09,000000 PM	modifPersona	Lluís, Hernández, Pérez, 38444001E, Pau Claris, 18, , 93444001, lmp@hotmail.com	ERROR: Hi ha camps que no poden ser nuls
221	04-JUN-11 10.02.09,000000 PM	modifPersona	Lluís, Martínez, Pérez, 54444001E, Pau Claris, 18, 948, 93444001, lmp@hotmail.com	ERROR: Es fa referència a una persona que no existeix a la base de dades o ha estat donada de baixa
222	04-JUN-11 10.02.09,000000 PM	modifPersona	Lluís, Hernández, Pérez, 38444001E, Pau Claris, 18, 990, 93444001, lmp@hotmail.com	ERROR: El municipi indicat no existeix
223	04-JUN-11 10.02.09,000000 PM	modifAlumne	4, , 2009, Àlex, Neus	ERROR: Hi ha camps que no poden ser nuls
224	04-JUN-11 10.02.09,000000 PM	modifAlumne	9, 23/10/97, 2009, Àlex, Neus	ERROR: L'identificador introduït no correspon a cap alumne
225	04-JUN-11 10.02.09,000000 PM	modifProfessor	9	ERROR: Hi ha camps que no poden ser nuls
226	04-JUN-11 10.02.09,000000 PM	modifProfessor	1, Filologia Catalana	ERROR: L'identificador introduït no correspon a cap professor
227	04-JUN-11 10.02.09,000000 PM	modifAssignatura	5, , Història	ERROR: Hi ha camps que no poden ser nuls
228	04-JUN-11 10.02.09,000000 PM	modifAssignatura	15, 3, Història	ERROR: L'assignatura indicada no existeix
229	04-JUN-11 10.02.09,000000 PM	modifAssignatura	1, 3, Matemàtiques	ERROR: Aquesta assignatura ja existeix
230	04-JUN-11 10.02.09,000000 PM	modifCurs	5, ,	ERROR: Hi ha camps que no poden ser nuls
231	04-JUN-11 10.02.09,000000 PM	modifCurs	15, 2n. Batxillerat, 2	ERROR: L'identificador introduït no correspon a cap curs
232	04-JUN-11 10.02.09,000000 PM	modifCurs	4, 4t. ESO, 2	ERROR: Aquest curs ja existeix
233	04-JUN-11 10.02.09,000000 PM	modifCurs	5, 1r. EGB, 2	ERROR: El nom del curs no és correcte
234	04-JUN-11 10.02.09,000000 PM	modifAmonestacio	4, Alumne_no_deures, L'alumne no fa els deures,	ERROR: Hi ha camps que no poden ser nuls

235	04-JUN-11 10.02.09,000000 PM	modifAmonestacio	14, Alumne_no_deures, L'alumne no fa els deures, Greu	ERROR: L'identificador indicat no correspon a cap amonestació
236	04-JUN-11 10.02.09,000000 PM	modifAmonestacio	1, Arriba_tard, L'alumne arriba tard a classe, Lleu	ERROR: El tipus d'amonestació ja existeix
237	04-JUN-11 10.02.09,000000 PM	modifAmonestacio	4, Alumne_no_deures, L'alumne no fa els deures, Mitjà	ERROR: La gravetat de l'amonestació només pot ser Lleu, Greu o Molt greu
238	04-JUN-11 10.02.09,000000 PM	modifSancio	1, , Automàticament quan l'alumne arriba tard tres vegades	ERROR: Hi ha camps que no poden ser nuls
239	04-JUN-11 10.02.09,000000 PM	modifSancio	15, Quedar-se dues hores extres d'estudi durant una setmana, Automàticament quan l'alumne arriba tard tres vegades	ERROR: L'identificador indicat no correspon a cap sanció
240	04-JUN-11 10.02.09,000000 PM	modifSancio	4, Personalitzada: repetir treball, L'alumne no ha treballat prou la feina requerida	ERROR: El tipus de sanció ja existeix
241	04-JUN-11 10.02.09,000000 PM	modifCalendariEscolar	7, calendari escolar 2010-2011, 10/01/11, , 1, 2, 13	ERROR: Hi ha camps que no poden ser nuls
242	04-JUN-11 10.02.09,000000 PM	modifCalendariEscolar	17, calendari escolar 2010-2011, 10/01/11, 6, 1, 2, 13	ERROR: No existeix cap registre a Calendari Escolar amb aquest identificador
243	04-JUN-11 10.02.09,000000 PM	modifCalendariEscolar	7, calendari escolar 2010-2011, 07/03/11, 6, 1, 2, 13	ERROR: La data indicada correspon a un dia festiu
244	04-JUN-11 10.02.09,000000 PM	modifCalendariEscolar	7, calendari escolar 2010-2011, 10/01/11, 6, 9, 2, 13	ERROR: L'identificador del curs i/o de l'assignatura introduït no existeix
245	04-JUN-11 10.02.09,000000 PM	modifCalendariEscolar	7, calendari escolar 2010-2011, 10/01/11, 5, 1, 2, 21	ERROR: L'hora i/o dia indicat no existeix
246	04-JUN-11 10.02.09,000000 PM	modifCalendariEscolar	6, calendari escolar 2010-2011, 10/01/11, 4, 2, 3, 12	ERROR: El registre de Calendari Escolar indicat ja existeix
247	04-JUN-11 10.02.09,000000 PM	llistatAlumnesCurs	1.Eso	ERROR: El nom del curs no és correcte
248	04-JUN-11 10.02.09,000000 PM	llistatAmonestacionsSancions	Amonestació	ERROR: La selecció del llistat no és correcta
249	04-JUN-11 10.02.09,000000 PM	llistatAmonSancionsAlumne	Amonestació, 1	ERROR: La selecció del llistat no és correcta
250	04-JUN-11 10.02.09,000000 PM	llistatAmonSancionsAlumne	Sancions, 20	ERROR: L'identificador introduït no correspon a cap alumne

(índex)

13.2. Taules

```
CREATE TABLE Provincia (
idProvincia NumberCONSTRAINTPK_Provincia PRIMARY KEY,
nomProvincia Varchar2(30) CONSTRAINT NN_nomProvincia NOT NULL
CONSTRAINT "CH_PROVINCIA" CHECK (nomProvincia IN
('Barcelona','Tarragona','Lleida','Girona')) ,
CONSTRAINT UN_nomProvincia UNIQUE(nomProvincia)
);

CREATE TABLE Comarca (
idComarca NumberCONSTRAINT PK_Comarca PRIMARY KEY,
nomComarca Varchar2(30) CONSTRAINT NN_nomComarca NOT NULL,
provincia NumberCONSTRAINT FK_provincia REFERENCES
Provincia(idProvincia) NOT NULL,
CONSTRAINT UN_nomComarca UNIQUE(nomComarca)
);

CREATE TABLE Municipi (
idMunicipi Number CONSTRAINT PK_Municipi PRIMARY KEY,
nomMunicipi Varchar2(50) CONSTRAINT NN_nomMunicipi NOT NULL,
codiPostal CHAR(5) CONSTRAINT NN_cPostal NOT NULL,
comarca Number CONSTRAINT FK_Comarca REFERENCES Comarca(idComarca) NOT
NULL
);

CREATE TABLE CentreEscolar (
idCentre Number CONSTRAINT PK_Centre PRIMARY KEY,
nomCentre Varchar2(50) CONSTRAINT NN_nomCentre NOT NULL,
cif Varchar2(10) CONSTRAINT NN_cif NOT NULL,
adreça Varchar2(50) CONSTRAINT NN_adreçaCentre NOT NULL,
municipi Number CONSTRAINT FK_ce REFERENCES Municipi(idMunicipi) NOT
NULL,
telèfon Number(10) CONSTRAINT NN_telefonCentre NOT NULL,
fax Number(10) CONSTRAINT NN_fax NOT NULL,
eMail Varchar2(30),
CONSTRAINT UN_cif UNIQUE(cif)
);

CREATE TABLE Persona (
idPersona Number CONSTRAINT PK_Persona PRIMARY KEY,
nom Varchar2(30) CONSTRAINT NN_nom NOT NULL,
cognom1 Varchar2(30) CONSTRAINT NN_cognom1 NOT NULL,
cognom2 Varchar2(30),
docIdentificatiu Varchar2(10) CONSTRAINT NN_docIdentificatiu NOT NULL,
adreça Varchar2(50) CONSTRAINT NN_adreça NOT NULL,
municipi Number CONSTRAINT FK1_per REFERENCES Municipi(idMunicipi) NOT
NULL,
telèfon Number(10) CONSTRAINT NN_telefon NOT NULL,
eMail Varchar2(30),
CONSTRAINT UN_docIdentificatiu UNIQUE(docIdentificatiu)
);

CREATE TABLE Alumne (
idPersona Number CONSTRAINT FK_Alumne REFERENCES Persona(idPersona),
numExpedient Varchar2(10) CONSTRAINT NN_numExpedient NOT NULL,
dataNaixement Date CONSTRAINT NN_dataNaixement NOT NULL,
anyIngres Char(4) CONSTRAINT NN_anyIngres NOT NULL,
```

```
nomPare Varchar2(30) CONSTRAINT NN_nomPare NOT NULL,
nomMare Varchar2(30) CONSTRAINT NN_nomMare NOT NULL,
CONSTRAINT PK_Alumne PRIMARY KEY(idPersona),
CONSTRAINT UN_expedient UNIQUE(numExpedient)
);

CREATE TABLE Professor (
idPersona Number CONSTRAINT FK_Professor REFERENCES Persona(idPersona),
titulacio Varchar2(30)CONSTRAINT NN_titulacio NOT NULL,
CONSTRAINT PK_Professor PRIMARY KEY(idPersona)
);

CREATE TABLE Curs (
idCurs Number CONSTRAINT PK_Curs PRIMARY KEY,
nomCurs Varchar2(30) CONSTRAINT NN_nomCurs NOT NULL
CONSTRAINT "CH_CURS" CHECK (nomCurs IN ('1r. ESO','2n. ESO','3r.
ESO','4t. ESO', '1r. Batxillerat', '2n. Batxillerat')),
nombreGrups CONSTRAINT NN_nGrups NOT NULL,
CONSTRAINT UN_nomCurs UNIQUE(nomCurs)
);

CREATE TABLE Assignatura (
idCurs Number CONSTRAINT FK_curs REFERENCES Curs(idCurs),
idAssignatura Number CONSTRAINT NN_assignatura NOT NULL,
nomAssignatura Varchar2(30) CONSTRAINT NN_nomAssignatura NOT NULL,
CONSTRAINT PK_assignatura PRIMARY KEY (idCurs, idAssignatura)
);

CREATE TABLE Amonestacio (
idAmonestacio Number CONSTRAINT PK_Amonestacio PRIMARY KEY,
tipus Varchar2 (40) CONSTRAINT NN_tipus NOT NULL,
descripcio Varchar2(100) CONSTRAINT NN_descripcio NOT NULL,
gravetat Varchar2(15) CONSTRAINT NN_gravetat NOT NULL,
CONSTRAINT CH_gravetat CHECK (gravetat in ('Lleu','Greu','Molt greu')),
CONSTRAINT UN_tipusAmon UNIQUE (tipus)
);

CREATE TABLE Sancio (
idSancio Number CONSTRAINT PK_Sancio PRIMARY KEY,
tipus Varchar2(100) CONSTRAINT NN1_tipus NOT NULL,
activacio Varchar2(100) CONSTRAINT NN_activacio NOT NULL,
CONSTRAINT UN_tipusSancio UNIQUE (tipus)
);

CREATE TABLE ReglaActivacio (
amonestacio Number CONSTRAINT FK1_regla REFERENCES
Amonestacio(idAmonestacio) NOT NULL,
operador Char(1) CONSTRAINT NN1_regla NOT NULL,
nombreAmonestacions Number CONSTRAINT NN2_regla NOT NULL,
sancio Number CONSTRAINT FK2_regla REFERENCES Sancio(idSancio)NOT NULL,
CONSTRAINT PK_regla PRIMARY KEY (amonestacio, sancio)
);

CREATE TABLE Acumulat (
idAlumne Number,
idAmonestacio Number CONSTRAINT FK1_acum REFERENCES
Amonestacio(idAmonestacio) NOT NULL,
nombreAcumulat Number DEFAULT 0 CONSTRAINT NN_acum NOT NULL,
```

```
CONSTRAINT PK_acum PRIMARY KEY (idAlumne, idAmonestacio)
);

CREATE TABLE Dates (
data DATE CONSTRAINT PK_dies PRIMARY KEY,
festiu CHAR(1 CHAR) CONSTRAINT NN_festiu NOT NULL,
CONSTRAINT CH_festiu CHECK (festiu in ('N','S'))
);

CREATE TABLE diaSetmana (
idDia Number CONSTRAINT PK_diaSetmana PRIMARY KEY,
nomDia Varchar2(10) CONSTRAINT NN_nomDia NOT NULL
CONSTRAINT CH_nomDia CHECK (nomDia in
('Dilluns','Dimarts','Dimecres','Dijous','Divendres'))
);

CREATE TABLE HoraLectiva (
idDia Number CONSTRAINT FK_dia REFERENCES DiaSetmana(idDia),
horaDia Number(2) CONSTRAINT NN_hora NOT NULL
CONSTRAINT CH_Hora CHECK (horaDia BETWEEN 08 AND 18),
CONSTRAINT PK_HoraLectiva PRIMARY KEY (idDia, horaDia)
);

CREATE TABLE CalendariEscolar(
idCalendari Number CONSTRAINT PK_calendari PRIMARY KEY,
nomCalendari Varchar2(35) CONSTRAINT NN_nomCalendari NOT NULL,
data DATE CONSTRAINT FK1_calendari REFERENCES Dates(data) NOT NULL,
curs Number CONSTRAINT NN_cursCalendari NOT NULL,
assignatura Number CONSTRAINT NN_assigCalendari NOT NULL,
dia Number CONSTRAINT NN_diaCalendari NOT NULL,
hora Number(2,0) CONSTRAINT NN_horaCalendari NOT NULL,
CONSTRAINT FK2_calendari FOREIGN KEY(curs, assignatura) REFERENCES
Assignatura(idCurs, idAssignatura),
CONSTRAINT FK3_calendari FOREIGN KEY(dia, hora) REFERENCES
HoraLectiva(idDia, horaDia),
CONSTRAINT UN_Calendari UNIQUE (data, curs, assignatura, dia, hora)
);

CREATE TABLE Es_tutor (
professor Number CONSTRAINT FK1_tutor REFERENCES Professor(idPersona),
curs Number CONSTRAINT FK2_tutor REFERENCES Curs(idCurs),
diaTutoria Number CONSTRAINT NN_diaTutoria NOT NULL,
horaIniciTutoria Number(2,0) CONSTRAINT NN_horaTutoria NOT NULL,
horaFiTutoria Number (2,0) CONSTRAINT NN_horaFi NOT NULL,
diaPares Number CONSTRAINT NN_DIAPARES NOT NULL,
horaPares Number(2,0) CONSTRAINT NN_HORAPARES NOT NULL,
CONSTRAINT FK3_tutor FOREIGN KEY(diaTutoria, horaIniciTutoria)
REFERENCES HoraLectiva(idDia, horaDia),
CONSTRAINT CH_tutor CHECK (horaFiTutoria > horaIniciTutoria),
CONSTRAINT PK_tutor PRIMARY KEY (professor, curs, diaTutoria,
horaIniciTutoria)
);

CREATE TABLE Matriculat (
alumne Number CONSTRAINT FK1_matriculat REFERENCES Alumne(idPersona),
curs Number CONSTRAINT FK2_matriculat REFERENCES Curs(idCurs),
centre Number CONSTRAINT FK3_matriculat REFERENCES
CentreEscolar(idCentre) NOT NULL,
CONSTRAINT PK_matriculat PRIMARY KEY (alumne, curs),
```

```
CONSTRAINT UN_matriculat UNIQUE (alumne, centre)
);

CREATE TABLE Imparteix (
professor Number CONSTRAINT FK1_imparteix REFERENCES
Professor(idPersona),
curs Number,
assignatura Number,
centre Number CONSTRAINT FK2_imparteix REFERENCES
CentreEscolar(idCentre) NOT NULL,
CONSTRAINT FK3_imparteix FOREIGN KEY(curs, assignatura) REFERENCES
Assignatura(idCurs, idAssignatura),
CONSTRAINT PK_imparteix PRIMARY KEY (professor, curs, assignatura),
CONSTRAINT UN_imparteix UNIQUE (professor, curs, assignatura, centre)
);

CREATE TABLE AmonestacionsEmeses(
Alumne Number CONSTRAINT FK1_amonEmeses REFERENCES Alumne(idPersona),
Professor Number CONSTRAINT FK2_amonEmeses REFERENCES
Professor(idPersona),
Amonestacio Number CONSTRAINT FK3_amonEmeses REFERENCES
Amonestacio(idAmonestacio) CONSTRAINT NN1_amonEmeses NOT NULL,
curs Number CONSTRAINT NN2_amonEmeses NOT NULL,
assignatura Number CONSTRAINT NN3_amonEmeses NOT NULL,
data DATE CONSTRAINT NN4_amonEmeses NOT NULL
CONSTRAINT FK4_amonEmeses REFERENCES Dates(data),
horaAmonestacio CHAR(5),
comunicacioPares CHAR (1 CHAR) CONSTRAINT NN_comunicat NOT NULL,
CONSTRAINT FK5_amonEmeses FOREIGN KEY(curs, assignatura) REFERENCES
Assignatura(idCurs, idAssignatura),
CONSTRAINT CH_comunicat CHECK (comunicacioPares in ('N','S')),
CONSTRAINT PK_Amonestacions_emeses PRIMARY KEY (alumne, professor,
amonestacio, data, horaAmonestacio)
);

CREATE TABLE SancionsEmeses(
alumne Number CONSTRAINT FK1_sancionsEmeses REFERENCES
Alumne(idPersona),
professor Number CONSTRAINT FK2_sancionsEmeses REFERENCES Professor
(idPersona),
sancio Number CONSTRAINT FK3_sancionsEmeses REFERENCES Sancio(idSancio),
data DATE CONSTRAINT NN_sancionsEmeses NOT NULL
CONSTRAINT FK4_sancionsEmeses REFERENCES Dates(data),
CONSTRAINT PK_Sancions_emeses PRIMARY KEY (alumne, professor, sancio,
data)
);

CREATE TABLE log(
idLog Number CONSTRAINT PK_log PRIMARY KEY,
data timestamp CONSTRAINT NN_data NOT NULL,
procediment Varchar2(50) CONSTRAINT NN_procediment NOT NULL,
entrada Varchar2(300) CONSTRAINT NN_entrada NOT NULL,
sortida Varchar2(200) CONSTRAINT NN_sortida NOT NULL
);

CREATE TABLE AmonestacionsAlumne (
idAlumne Number CONSTRAINT PK_amonAlumne PRIMARY KEY,
nomAlumne Varchar2(60) CONSTRAINT NN_amonAlumne NOT NULL,
```

```
nombreAmonAlumne Number CONSTRAINT NN_nombreAmon NOT NULL,
observacions Varchar2(40)
);

CREATE TABLE SancionsAlumne (
idAlumne Number,
nomAlumne Varchar2(60) CONSTRAINT NN1_sa_any NOT NULL,
idAny Char(4) CONSTRAINT NN2_sa_any NOT NULL,
nombreSancionsAlumne Number CONSTRAINT NN_nombreSanAl NOT NULL,
observacions Varchar2(40),
CONSTRAINT PK_sanAlumne PRIMARY KEY(idAlumne, idAny)
);

CREATE TABLE SancionsCursAny (
idCurs Number,
nomCurs Varchar2(20) CONSTRAINT NN_sa_curs NOT NULL,
idAny Char(4),
nombreSancionsCurs Number CONSTRAINT NN_nombreSan NOT NULL,
observacions Varchar2(40),
CONSTRAINT PK_sanCurs PRIMARY KEY(idCurs, idAny)
);

CREATE TABLE AvgAmonestacionsProfAny (
idProfessor Number CONSTRAINT PK_avgapa PRIMARY KEY,
nomProfessor Varchar2(60) CONSTRAINT NN_avgprofessor NOT NULL,
avgAmonProfAny Number CONSTRAINT NN_avgapa NOT NULL,
observacions Varchar2(40)
);

CREATE TABLE AvgSancionsCurs (
idCurs Number CONSTRAINT PK_avgsaa PRIMARY KEY,
nomCurs Varchar2(20) CONSTRAINT NN_avg_curs NOT NULL,
avgSancionsCurs Number CONSTRAINT NN1_avgsaa NOT NULL,
observacions Varchar2(40)
);

CREATE TABLE AlumneMesSancionat (
idAlumne Number,
nomAlumne Varchar2(60) CONSTRAINT NN_amsAlumne NOT NULL,
idAny Char(4) CONSTRAINT NN_amsAny NOT NULL,
observacions Varchar2(40),
CONSTRAINT PK_ams PRIMARY KEY (idAlumne, idAny)
);

CREATE TABLE ProfessorMesAmonestador (
idProfessor Number,
nomProfessor Varchar2(60) CONSTRAINT NN_pmaProf NOT NULL,
idCurs Number CONSTRAINT NN1_pmsCurs NOT NULL,
nomCurs Varchar2(20) CONSTRAINT NN2_pmaCurs NOT NULL,
observacions Varchar2(40),
CONSTRAINT PK_pma PRIMARY KEY(idProfessor, idCurs)
);

CREATE TABLE AlumnesNoAmonestats (
nombreAlNoAmon Number CONSTRAINT PK_ana PRIMARY KEY
);
```

(índex)

13.3. Seqüències

```
CREATE SEQUENCE seq_Provincia INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idProvincia_Provincia
BEFORE INSERT ON Provincia
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Provincia.NEXTVAL, 6, 0) INTO :NEW.idProvincia
FROM DUAL;
END inserir_idProvincia_Provincia;

CREATE SEQUENCE seq_Comarca INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idComarca_Comarca
BEFORE INSERT ON Comarca
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Comarca.NEXTVAL, 6, 0) INTO :NEW.idComarca
FROM DUAL;
END inserir_idComarca_Comarca;

CREATE SEQUENCE seq_Municipi INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idMunicipi_Municipi
BEFORE INSERT ON Municipi
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Municipi.NEXTVAL, 6, 0) INTO :NEW.idMunicipi
FROM DUAL;
END inserir_idMunicipi_Municipi;

CREATE SEQUENCE seq_CentreEscolar INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idCentre_Centre
BEFORE INSERT ON CentreEscolar
FOR EACH ROW
BEGIN
SELECT LPAD(seq_CentreEscolar.NEXTVAL, 6, 0) INTO :NEW.idCentre
FROM DUAL;
END inserir_idCentre_Centre;

CREATE SEQUENCE seq_Persona INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idPersona_Persona
BEFORE INSERT ON Persona
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Persona.NEXTVAL, 6, 0) INTO :NEW.idPersona
FROM DUAL;
END inserir_idPersona_Persona;

CREATE SEQUENCE seq_Curs INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idCurs_Curs
BEFORE INSERT ON Curs
```

```
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Curs.NEXTVAL, 6, 0) INTO :NEW.idCurs
FROM DUAL;
END inserir_idCurs_Curs;

CREATE SEQUENCE seq_Amon INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idAmon_Amon
BEFORE INSERT ON Amonestacio
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Amon.NEXTVAL, 6, 0) INTO :NEW.idAmonestacio
FROM DUAL;
END inserir_idAmon_Amon;

CREATE SEQUENCE seq_Sancio INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idSancio_Sancio
BEFORE INSERT ON Sancio
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Sancio.NEXTVAL, 6, 0) INTO :NEW.idSancio
FROM DUAL;
END inserir_idSancio_Sancio;

CREATE SEQUENCE seq_DiaSetmana INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idDia_Dia
BEFORE INSERT ON DiaSetmana
FOR EACH ROW
BEGIN
SELECT LPAD(seq_DiaSetmana.NEXTVAL, 6, 0) INTO :NEW.idDia
FROM DUAL;
END inserir_idDia_Dia;

CREATE SEQUENCE seq_Calendari INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idCalendari
BEFORE INSERT ON CalendariEscolar
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Calendari.NEXTVAL, 6, 0) INTO :NEW.idCalendari
FROM DUAL;
END inserir_idCalendari;

CREATE SEQUENCE seq_Log INCREMENT BY 1 START WITH 1;
CREATE OR REPLACE
TRIGGER inserir_idLog_Log
BEFORE INSERT ON Log
FOR EACH ROW
BEGIN
SELECT LPAD(seq_Log.NEXTVAL, 6, 0) INTO :NEW.idLog
FROM DUAL;
END inserir_idLog_Log;
```

(índex)

13.4. Funcions

```
/*****************************************************************
Funció auxiliar per obtenir el nom d'un alumne
Nom funció: getNomAlumne
Paràmetres d'entrada: f_idAlumne
Sortida: RSP
Retorna: Retorna el nom de l'alumne. En el Log, "OK" si l'execució
finalitza amb èxit; "ERROR: Tipus d'error" si fracassa.
*****************************************************************/

CREATE OR REPLACE FUNCTION getNomAlumne (f_idAlumne NUMBER)
RETURN CHAR
IS
  nomAlumne  Varchar2(60);
  RSP        Varchar2(100);
  alumneNul  EXCEPTION;
BEGIN
/*Es verifica que s'introdueixi l'identificador de l'alumne*/
  IF (f_idAlumne IS NULL)
  THEN
    RAISE alumneNul;
  END IF;
/*S'obté el nom de l'alumne*/
  SELECT distinct(p.nom ||' ' || p.cognom1||' ' || p.cognom2) INTO
nomAlumne
  FROM Persona p, Alumne al
  WHERE p.idPersona = al.idPersona AND al.idPersona = f_idAlumne;
/*Es retorna el nom de l'alumne */
  RETURN nomAlumne;
  EXCEPTION
  WHEN alumneNul THEN
  RSP:= 'ERROR: l''identificador de l''alumne no pot ser nul';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'getNomAlumne', f_idAlumne, RSP);
END;
```

(índex)

```
/*****************************************************************
Funció auxiliar per obtenir el nom d'un professor.
Nom funció: getNomProfessor
Paràmetres d'entrada: f_idProfessor
Sortida: RSP
Retorna: Retorna el nom del professor requerit. En el Log, "OK" si
l'execució finalitza amb èxit; "ERROR: Tipus d'error" si fracassa.
*****************************************************************/


CREATE OR REPLACE FUNCTION getNomProfessor (f_idProfessor NUMBER)
RETURN CHAR
IS
  nomProfessor  Varchar2(60);
  RSP           Varchar2(100);
  professorNul  EXCEPTION;
BEGIN
/*Es verifica que s'introdueixi l'identificador del professor*/
  IF (f_idProfessor IS NULL)
  THEN
    RAISE professorNul;
  END IF;
/*S'obté el nom del professor*/
  SELECT distinct(p.nom ||' ' || p.cognom1||' ' || p.cognom2) INTO
nomProfessor
  FROM Persona p, Professor r
  WHERE p.idPersona = r.idPersona AND r.idPersona = f_idProfessor;
/*Es retorna el nom del professor*/
  RETURN nomProfessor;
  EXCEPTION
  WHEN professorNul THEN
  RSP:= 'ERROR: l''identificador del professor no pot ser nul';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'getNomProfessor', f_idProfessor, RSP);
END;
```

(índex)

```
/*********************************************************************
Funció auxiliar per obtenir el nom del curs a què pertany un alumne.
Nom funció: getNomCurs
Paràmetres d'entrada: f_idAlumne
Sortida: RSP
Retorna: Retorna el nom del curs requerit. En el Log, "OK" si l'execució
finalitza amb èxit; "ERROR: Tipus d'error" si fracassa.
*********************************************************************/


CREATE OR REPLACE FUNCTION getNomCurs (f_alumne NUMBER)
RETURN CHAR
IS
  f_nomCurs     Varchar2(30);
  RSP           Varchar2(100);
  cursNul       EXCEPTION;
BEGIN
/*Es verifica que s'introdueixi l'identificador de l'alumne*/
  IF (f_alumne IS NULL)
  THEN
    RAISE cursNul;
  END IF;
/*S'obté el nom del curs a què pertany l'alumne*/
  SELECT nomCurs INTO f_nomCurs
  FROM Curs, Matriculat, Alumne
  WHERE idCurs = curs AND alumne = idPersona AND idPersona = f_alumne;
/*Es retorna el nom del curs*/
  RETURN f_nomCurs;
  EXCEPTION
  WHEN cursNul THEN
  RSP:= 'ERROR: l''identificador de l''alumne no pot ser nul';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'getNomCurs', f_alumne, RSP);
END;
```

(índex)

```
/*************************************************************************
Funció auxiliar per obtenir l'operador definit en una regla d'activació
de sancions automàtiques
Nom funció: getOperadorRegla
Paràmetres d'entrada: f_idAmonestacio
Sortida: RSP
Retorna: Retorna l'operador definit en les regles d'activació. En el
Log, "OK" si l'execució finalitza amb èxit; "ERROR: Tipus d'error" si
fracassa.
*************************************************************************/


CREATE OR REPLACE FUNCTION getOperadorRegla (f_idAmonestacio NUMBER)
RETURN CHAR
IS
  f_operador           Char(1);
  RSP                  Varchar2(100);
  idAmonestacioNul     EXCEPTION;
BEGIN
/*Es verifica que s'introdueixi l'identificador de l'amonestacio*/
  IF (f_idAmonestacio IS NULL)
  THEN
    RAISE idAmonestacioNul;
  END IF;
/*S'obté l'operador*/
  SELECT operador INTO f_operador
  FROM ReglaActivacio ra, Amonestacio
  WHERE ra.amonestacio = idAmonestacio AND idAmonestacio =
f_idAmonestacio;
/*Es retorna l'operador */
  RETURN f_operador;
  EXCEPTION
  WHEN idAmonestacioNul THEN
  RSP:= 'ERROR: l''identificador de l''amonestació no pot ser nul';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'getOperadorRegla', f_idAmonestacio, RSP);
END;
```

(índex)

```
/*********************************************************************
Funció auxiliar per obtenir el nombre d'amonestacions d'un mateix tipus
definit en una regla d'activació de sancions automàtiques
Nom funció: getNombreAcumulatRegla
Paràmetres d'entrada: f_idAmonestacio
Sortida: RSP
Retorna: Retorna el nombre acumulat d'amonestacions definit en les
regles d'activació. En el Log, "OK" si l'execució finalitza amb èxit;
"ERROR: Tipus d'error" si fracassa.
*********************************************************************/

CREATE OR REPLACE FUNCTION getNombreAcumulatRegla (f_idAmonestacio
NUMBER)
RETURN NUMBER
IS
  f_nombreAcum          Char(1);
  RSP                   Varchar2(100);
  idAmonestacioNul      EXCEPTION;
BEGIN
/*Es verifica que s'introdueix l'identificador de l'amonestacio*/
  IF (f_idAmonestacio IS NULL)
  THEN
    RAISE idAmonestacioNul;
  END IF;
/*S'obté el nombre d'amonestacions del mateix tipus que cal tenir
acumulades per activar la sanció automàtica*/
  SELECT nombreAmonestacions INTO f_nombreAcum
  FROM ReglaActivacio ra, Amonestacio
  WHERE ra.amonestacio = idAmonestacio AND idAmonestacio =
f_idAmonestacio;
/*Es retorna el nombre acumulat d'amonestacions definit a la regla */
  RETURN f_nombreAcum;
  EXCEPTION
  WHEN idAmonestacioNul THEN
  RSP:= 'ERROR: l''identificador de l''amonestació no pot ser nul';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'getNombreAcumulatRegla', f_idAmonestacio, RSP);
END;
```

(índex)

```
/*************************************************************************
Funció auxiliar per obtenir la sanció automàtica definida en una regla
d'activació
Nom funció: getSancioAutomatica
Paràmetres d'entrada: f_idAmonestacio
Sortida: RSP
Retorna: Retorna l'identificador de la sanció automàtica que
correspongui definit en les regles d'activació. En el Log, "OK" si
l'execució finalitza amb èxit; "ERROR: Tipus d'error" si fracassa.
*************************************************************************/

CREATE OR REPLACE FUNCTION getSancioAutomatica (f_idAmonestacio NUMBER)
RETURN NUMBER
IS
  f_sancioAut          Char(1);
  RSP                  Varchar2(100);
  idAmonestacioNul     EXCEPTION;
BEGIN
/*Es verifica que s'introdueix l'identificador de l'amonestacio*/
  IF (f_idAmonestacio IS NULL)
  THEN
    RAISE idAmonestacioNul;
  END IF;
/*S'obté la sanció automàtica definida a la regla d'activació*/
  SELECT sancio INTO f_sancioAut
  FROM ReglaActivacio ra, Amonestacio
  WHERE ra.amonestacio = idAmonestacio AND idAmonestacio =
f_idAmonestacio;
/*Es retorna l'operador */
  RETURN f_sancioAut;
  EXCEPTION
  WHEN idAmonestacioNul THEN
  RSP:= 'ERROR: l''identificador de l''amonestació no pot ser nul';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'getSancioAutomatica', f_idAmonestacio, RSP);
END;
```

(índex)

13.5. Procediments emmagatzemats

```
/*********************************************************************
Procediment que gestiona l'actualització dels registres de la taula
estadística AmonestacionsAlumne que reflecteix el nombre d'amonestacions
per alumne, independentment del curs
Nom Procediment: estAmonestacionsAlumne
Paràmetres d'entrada: p_alumne
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/

CREATE OR REPLACE PROCEDURE estAmonestacionsAlumne (
  p_alumne IN Number,
  RSP OUT Varchar2
)
IS
  amNomAlumne Varchar2(60);
  hihaAmonAlumne Number;
  nombreAmAlumne Number;
  campsNuls EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_alumne IS NULL) THEN
    RAISE campsNuls;
  END IF;
  /*S'obté el nom de l'alumne*/
  amNomAlumne:= getNomAlumne (p_alumne);
  /*Es comprova si ja hi ha dades estadístiques*/
  SELECT COUNT(*) INTO hihaAmonAlumne
  FROM AmonestacionsAlumne
  WHERE idAlumne = p_alumne;
  /*Si no hi havia, s'hi afegeix*/
  IF (hihaAmonAlumne = 0) THEN
   INSERT     INTO     AmonestacionsAlumne     (idAlumne,     nomAlumne,
 nombreAmonAlumne)
    VALUES (p_alumne,amNomAlumne, 1);
  ELSE
  /*Si n'hi havia, s'obté el nombre d'amonestacions anteriors i es suma
    l'actual*/
   SELECT nombreAmonAlumne INTO nombreAmAlumne
   FROM AmonestacionsAlumne
   WHERE idAlumne = p_alumne;
   UPDATE AmonestacionsAlumne
   SET nombreAmonAlumne = nombreAmAlumne + 1
   WHERE idAlumne = p_alumne;
  END IF;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAmonestacionsAlumne', p_alumne, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
```

```
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAmonestacionsAlumne', p_alumne, RSP);
  WHEN others THEN
  RSP:=   'ERROR:   Error   genèric   en   actualitzar   l''estadística
    d''amonestacions per alumne';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAmonestacionsAlumne', p_alumne, RSP);
END estAmonestacionsAlumne;
```

(índex)

```
/*****************************************************************
Procediment que gestiona l'actualització dels registres de la taula
estadística SancionsAlumne que reflecteix el nombre de sancions per
alumne i any
Nom Procediment: estSancionsAlumne
Paràmetres d'entrada: p_alumne, p_data
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*****************************************************************/

CREATE OR REPLACE PROCEDURE estSancionsAlumne (
  p_alumne IN Number,
  p_data IN Date,
  RSP OUT Varchar2
)
IS
  sancioNomAlumne Varchar2(60);
  sancioAny Char(4);
  hihaSancioAlumne Number;
  nombreSanAlumne Number;
  campsNuls EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_alumne IS NULL)
    OR (p_data IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*S'obtenen les dades de l'alumne i l'any*/
  sancioNomAlumne:= getNomAlumne (p_alumne);
  sancioAny:= EXTRACT (YEAR FROM p_data);
 /*Es comprova si ja hi ha dades estadístiques*/
  SELECT COUNT(*) INTO hihaSancioAlumne
  FROM SancionsAlumne
  WHERE idAlumne = p_alumne AND idAny = sancioAny;
/*Si no hi havia, s'hi afegeix*/
  IF (hihaSancioAlumne = 0) THEN
    INSERT    INTO    SancionsAlumne    (idAlumne,    nomAlumne,    idAny,
 nombreSancionsAlumne)
      VALUES (p_alumne,sancioNomAlumne,sancioAny,1);
  ELSE
/*Si n'hi havia, s'obté el nombre de sancions anteriors i es suma
   l'actual*/
    SELECT nombreSancionsAlumne INTO nombreSanAlumne
    FROM SancionsAlumne
    WHERE idAlumne = p_alumne AND idAny = sancioAny;
    UPDATE SancionsAlumne
    SET    nombreSancionsAlumne = nombreSanAlumne + 1
    WHERE idAlumne = p_alumne AND idAny = sancioAny;
  END IF;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estSancionsAlumne', p_alumne || ', ' || p_data,
    RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
```

```
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estSancionsAlumne', p_alumne || ', ' || p_data,
    RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric actualitzar l''estadística de sancions per
    alumne i any';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estSancionsAlumne', p_alumne || ', ' || p_data,
    RSP);
END estSancionsAlumne;
```

(índex)

```
/*********************************************************************
Procediment que gestiona l'actualització dels registres de la taula
estadística SancionsCursAny que reflecteix el nombre de sancions per
curs i any
Nom Procediment: estSancionsCursAny
Paràmetres d'entrada: p_alumne, p_data
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/

CREATE OR REPLACE PROCEDURE estSancionsCursAny (
  p_alumne IN Number,
  p_data IN Date,
  RSP OUT Varchar2
)
IS
  sancioNomCurs Varchar2(20);
  sIdCurs Number;
  sancioAny Char(4);
  nombreSanCurs Number;
  hihaSancioCurs Number;
  campsNuls EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_alumne IS NULL)
    OR (p_data IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*S'obté el nom del curs i l'any*/
  sancioNomCurs:= getNomCurs (p_alumne);
  sancioAny:= EXTRACT (YEAR FROM p_data);
/*S'obté l'identificador del curs*/
  SELECT m.curs INTO sIdCurs
  FROM Alumne, Matriculat m
  WHERE idPersona = m.alumne AND idPersona = p_alumne;
/*Es comprova si ja hi ha dades estadístiques*/
  SELECT COUNT(*) INTO hihaSancioCurs
  FROM SancionsCursAny
  WHERE idCurs = sIdCurs AND idAny = sancioAny;
/*Si no hi havia, s'hi afegeix*/
  IF (hihaSancioCurs = 0)THEN
    INSERT    INTO    SancionsCursAny    (idCurs,    nomCurs,    idAny,
    nombreSancionsCurs)
      VALUES (sIdCurs,sancioNomCurs,sancioAny,1);
  ELSE
/*Si n'hi havia, s'obté el nombre de sancions anteriors i es suma
    l'actual*/
  SELECT nombreSancionsCurs INTO nombreSanCurs
  FROM SancionsCursAny
  WHERE idCurs = sIdCurs AND idAny = sancioAny;
  UPDATE SancionsCursAny
  SET  nombreSancionsCurs = nombreSanCurs + 1
  WHERE idCurs = sIdCurs AND idAny = sancioAny;
  END IF;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
```

```
      VALUES (SYSDATE, 'estSancionsCursAny', p_alumne || ', ' || p_data,
    RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estSancionsCursAny', p_alumne || ', ' || p_data,
    RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en actualitzar l''estadística de sancions
    per curs i any';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estSancionsCursAny', p_alumne || ', ' || p_data,
    RSP);
END estSancionsCursAny;
```

(índex)

```
/*********************************************************************
Procediment que gestiona l'actualització dels registres a la taula
estadística AvgAmonestacionsProfAny que reflecteix la mitjana
d'amonestacions per professor i any.
Nom Procediment: estAvgAmonProfAny
Paràmetres d'entrada: p_professor, p_data
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/

CREATE OR REPLACE PROCEDURE estAvgAmonProfAny (
  p_professor IN Number,
  p_data IN Date,
  RSP OUT Varchar2
)
IS
  amNomProfessor Varchar2(60);
  amAny Char(4);
  hihaAvgProf Number;
  mitjana Number;
  campsNuls EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_professor IS NULL)
    OR (p_data IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es calcula la mitjana*/
  SELECT avg(total) INTO mitjana
  FROM(SELECT professor, EXTRACT(YEAR FROM data), COUNT(amonestacio) as
        total
        FROM AmonestacionsEmeses
        WHERE professor = p_professor
        GROUP BY professor, EXTRACT(YEAR FROM data));
/*Es troba el nom del professor*/
  amNomProfessor:= getNomProfessor(p_professor);
/*Es comprova si ja hi ha dades estadístiques*/
  SELECT COUNT(*) INTO hihaAvgProf
  FROM AvgAmonestacionsProfAny
  WHERE idProfessor = p_professor;
 /*Si no hi havia, s'hi afegeix*/
  IF (hihaAvgProf = 0) THEN
    INSERT INTO AvgAmonestacionsProfAny (idProfessor, nomProfessor,
 avgAmonProfAny)
      VALUES (p_professor, amNomProfessor, mitjana);
  ELSE
  /*Si n'hi havia, s'actualitza*/
    UPDATE AvgAmonestacionsProfAny
    SET      avgAmonProfAny = mitjana
    WHERE idProfessor = p_professor;
  END IF;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAvgAmonProfAny', p_professor || ', ' || p_data,
    RSP);
  COMMIT;
```

```
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAvgAmonProfAny', p_professor || ', ' || p_data,
    RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en actualitzar l''estadística de mitjana
    d''amonestacions per professor i any';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAvgAmonProfAny', p_professor || ', ' || p_data,
    RSP);
END estAvgAmonProfAny;
```

(índex)

```
/*********************************************************************
Procediment que gestiona l'actualització dels registres a la taula
estadística AvgSancionsCurs que reflecteix la mitjana de sancions que
tenen els alumnes de cada curs.
Nom Procediment: estAvgSancionsCurs
Paràmetres d'entrada: p_alumne
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/

CREATE OR REPLACE PROCEDURE estAvgSancionsCurs (
  p_alumne IN Number,
  RSP OUT Varchar2
)
IS
  sancioNomCurs Varchar2(20);
  sIdCurs Number;
  hihaAvgCurs Number;
  mitjana Number;
  campsNuls EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_alumne IS NULL) THEN
    RAISE campsNuls;
  END IF;
  /*Es calcula la mitjana*/
  SELECT avg(total) INTO mitjana
  FROM(SELECT m.curs, se.alumne, COUNT(se.sancio) as total
        FROM  SancionsEmeses se, Matriculat m
        WHERE se.alumne = m.alumne
        GROUP BY m.curs, se.alumne);
 /*S'obté el nom del curs*/
  sancioNomCurs:= getNomCurs (p_alumne);
  /*S'obté l'identificador del curs*/
  SELECT m.curs INTO sIdCurs
  FROM Alumne, Matriculat m
  WHERE idPersona = m.alumne AND idPersona = p_alumne;
 /*Es comprova si ja hi ha dades estadístiques*/
  SELECT COUNT(*) INTO hihaAvgCurs
  FROM AvgSancionsCurs
  WHERE idCurs IN (SELECT idCurs
                   FROM Matriculat m, Alumne
                   WHERE   m.alumne   =   idPersona   AND   idPersona   =
                   p_alumne);
 /*Si no hi havia cap dada, s'hi afegeix*/
  IF (hihaAvgCurs = 0) THEN
    INSERT INTO AvgSancionsCurs (idCurs, nomCurs, avgSancionsCurs)
       VALUES (sIdCurs, sancioNomCurs, mitjana);
  ELSE
  /*Si n'hi havia, s'actualitza*/
    UPDATE AvgSancionsCurs
    SET avgSancionsCurs = mitjana
    WHERE idCurs = sIdCurs;
  END IF;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
       VALUES (SYSDATE, 'estAvgSancionsCurs', p_alumne, RSP);
```

```
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAvgSancionsCurs', p_alumne, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en actualitzar l''estadística de mitjana
    de sancions per cada curs';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAvgSancionsCurs', p_alumne, RSP);
END estAvgSancionsCurs;
```

(índex)

```
/*****************************************************************
Procediment que gestiona l'actualització de les dades a la taula
estadística AlumneMesSancionat que retorna el nom de l'alumne que ha
rebut més sancions en un any concret
Nom Procediment: estAlumneMesSancionat
Paràmetres d'entrada: p_alumne, p_data
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*****************************************************************/

CREATE OR REPLACE PROCEDURE estAlumneMesSancionat (
  p_alumne IN Number,
  p_data IN Date,
  RSP OUT Varchar2
)
IS
  hihaAlumneMesSan Number;
  sancioNomAlumne Varchar2(60);
  alumneMesSan Number;
  campsNuls EXCEPTION;
  nombreFiles Number;
  CURSOR c_alMesSan IS
  SELECT alumne
  FROM SancionsEmeses
  WHERE EXTRACT(YEAR FROM data) = EXTRACT(YEAR FROM p_data)
  GROUP BY alumne
  HAVING COUNT(alumne) =(SELECT MAX(totalSancio)
                         FROM(SELECT alumne, count(alumne) as totalSancio
                              FROM SancionsEmeses
                              WHERE    EXTRACT(YEAR    FROM    data)    =
                              EXTRACT(YEAR FROM p_data)
                              GROUP BY alumne));
  cur_alMesSan           c_alMesSan%ROWTYPE;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_alumne IS NULL)
    OR (p_data IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
  /*S'obté el nom de l'alumne*/
  sancioNomAlumne:= getNomAlumne (p_alumne);
/*Es comprova si ja hi ha dades estadístiques per a aquell any*/
  SELECT COUNT(*) INTO hihaAlumneMesSan
  FROM AlumneMesSancionat
  WHERE idAny = EXTRACT(YEAR FROM p_data);
/*Si no hi havia, s'hi afegeix*/
  IF (hihaAlumneMesSan = 0)THEN
  INSERT INTO AlumneMesSancionat (idAlumne, nomAlumne, idAny)
    VALUES (p_alumne, sancioNomAlumne, EXTRACT(YEAR FROM p_data));
  ELSE
/*Si n'hi havia, s'actualitza. Primer, mitjançant el cursor, es comprova
  si hi ha més d'un alumne amb el mateix nombre de sancions màxim. Si
  n'hi ha més d'un, es mostra un missatge que ho indica a la taula de
  l'alumne més sancionat i també a la taula Log*/
  FOR cur_alMesSan IN c_alMesSan
   LOOP
       DBMS_OUTPUT.PUT_LINE (cur_alMesSan.alumne);
```

```
      nombreFiles:= c_alMesSan%rowcount;
      IF (nombreFiles>1) THEN
        UPDATE AlumneMesSancionat
        SET idAlumne = 9999,
        nomAlumne ='No hi ha un únic alumne amb un màxim de sancions'
        WHERE idAny = EXTRACT(YEAR FROM p_data);
      ELSE
        UPDATE AlumneMesSancionat
        SET idAlumne = cur_alMesSan.alumne,
            nomAlumne =getNomAlumne(cur_alMesSan.alumne)
        WHERE idAny = EXTRACT(YEAR FROM p_data);
      END IF;
   END LOOP;
  END IF;
  IF (nombreFiles > 1) THEN
    RSP:= 'No hi ha un únic alumne amb un màxim de sancions per a aquest
          any';
  ELSE
    RSP:= 'OK';
  END IF;
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAlumneMesSancionat', p_alumne || ', ' || p_data,
    RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAlumneMesSancionat', p_alumne || ', ' || p_data,
    RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en actualitzar l''estadística d''alumne
    més sancionat en un any';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAlumneMesSancionat', p_alumne || ', ' || p_data,
    RSP);
END estAlumneMesSancionat;
```

(índex)

```
/*****************************************************************
Procediment que gestiona l'actualització de les dades a la taula
estadística ProfessorMesAmonestador que retorna el nom del professor que
ha imposat més amonestacions per cada curs
Nom Procediment: estProfMesAmon
Paràmetres d'entrada: p_professor, p_curs
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*****************************************************************/

CREATE OR REPLACE PROCEDURE estProfMesAmon (
   p_professor IN Number,
   p_alumne IN Number,
   p_curs IN Number,
   RSP OUT Varchar2
)
IS
   amNomProfessor Varchar2(60);
   amCurs Varchar2(20);
   hihaProfMesAmon NUMBER;
   profMesAmon NUMBER;
   campsNuls EXCEPTION;
   nombreFiles NUMBER;
   CURSOR c_maximProf IS
      SELECT professor
      FROM AmonestacionsEmeses
      WHERE curs = p_curs
      GROUP BY professor
      HAVING COUNT(professor)=(SELECT MAX(totalAmon)
                               FROM (SELECT professor, count(professor)
                                     as totalAmon
                                     FROM AmonestacionsEmeses
                                     WHERE curs = p_curs
                                     GROUP BY professor));
   cur_maximProf      c_maximProf%ROWTYPE;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
   IF (p_professor IS NULL)
      OR (p_alumne IS NULL)
      OR (p_curs IS NULL)
   THEN
      RAISE campsNuls;
   END IF;
/*S'obté el nom del professor i del curs*/
   amNomProfessor:= getNomProfessor(p_professor);
   amCurs:= getNomCurs (p_alumne);
/*Es comprova si ja hi ha dades estadístiques per a aquest curs*/
   SELECT COUNT(*) INTO hihaProfMesAmon
   FROM ProfessorMesAmonestador
   WHERE idCurs = p_curs;
/*Si no hi havia, s'hi afegeix*/
   IF (hihaProfMesAmon = 0) THEN
      INSERT INTO ProfessorMesAmonestador (idProfessor, nomProfessor,
        idCurs, nomCurs)
        VALUES (p_professor, amNomProfessor, p_curs, amCurs);
   ELSE
/*Si n'hi havia, s'actualitza. Primer, mitjançant el cursor, es comprova
 si hi ha més d'un professor amb el mateix nombre d'amonestacions
```

```
 màxim. Si n'hi ha més d'un, es mostra un missatge que ho indica a la
 taula del professor més amonestat i també a la taula Log*/
    FOR cur_maximProf IN c_maximProf
      LOOP
        DBMS_OUTPUT.PUT_LINE (cur_maximProf.professor);
        nombreFiles:= c_maximProf%rowcount;
        IF (nombreFiles > 1) THEN
          UPDATE ProfessorMesAmonestador
          SET  idProfessor = 9999,
               nomProfessor = 'No hi ha un únic professor com a màxim
                               amonestador'
          WHERE idCurs = p_curs;
        ELSE
          UPDATE ProfessorMesAmonestador
          SET  idProfessor = cur_maximProf.professor,
               nomProfessor = getNomProfessor(cur_maximProf.professor)
          WHERE idCurs = p_curs;
        END IF;
      END LOOP;
  END IF;
  IF (nombreFiles > 1) THEN
    RSP:= 'No hi ha un únic professor com a màxim amonestador en aquest
           curs';
  ELSE
    RSP:= 'OK';
  END IF;
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estProfMesAmon', p_professor || ', ' || p_curs,
            RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estProfMesAmon', p_professor || ', ' || p_curs,
            RSP);
  WHEN others THEN
  RSP:=  'ERROR:  Error  genèric  en  actualitzar  l'estadística  del
          professor més amonestador';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estProfMesAmon', p_professor || ', ' || p_curs,
            RSP);
END estProfMesAmon;
```

(índex)

```
/*****************************************************************
Procediment que gestiona l'actualització de les dades a la taula
estadística AlumnesNoAmonestats que retorna el nombre total d'alumnes
que no han rebut cap amonestació, sense fer cap tipus de filtrat.
Nom Procediment: estAlumnesNoAmon
Paràmetres d'entrada: cap
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*****************************************************************/

CREATE OR REPLACE PROCEDURE estAlumnesNoAmon (
  RSP OUT Varchar2
)
IS
  noAmonestats Number;
BEGIN
/*Es calcula el nombre d'alumnes no amonestats*/
  SELECT COUNT(*) INTO noAmonestats
  FROM Alumne
  WHERE idPersona NOT IN (SELECT alumne
                          FROM AmonestacionsEmeses
                          WHERE idPersona = alumne);
/*I s'actualitza la dada a la taula estadística*/
  UPDATE AlumnesNoAmonestats
  SET    nombreAlNoAmon = noAmonestats;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAlumnesNoAmon', ' ', RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en actualitzar l''estadística d''alumnes
    no amonestats';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'estAlumnesNoAmon', ' ', RSP);
END estAlumnesNoAmon;
```

(índex)

```
/*********************************************************************

Procediment que gestiona l'alta d'una persona a la taula Persona
Nom Procediment: altaPersona
Paràmetres d'entrada: p_nom, p_cognom1, p_cognom2, p_docIdentificatiu,
p_adreça, p_codiPostal, p_telefon, p_email
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE altaPersona (
  p_nom IN Varchar2,
  p_cognom1 IN Varchar2,
  p_cognom2 IN Varchar2,
  p_docIdentificatiu IN Varchar2,
  p_adreça IN Varchar2,
  p_municipi IN Number,
  p_telefon IN Number,
  p_email IN Varchar2,
  RSP OUT Varchar2
)
IS
 docIdDuplicat Number;
 noMunicipi Number;
 campsNuls EXCEPTION;
 docIdentificatiuDuplicat EXCEPTION;
 noExisteixMunicipi EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
 IF(p_nom IS NULL)
   OR (p_cognom1 IS NULL)
   OR (p_docIdentificatiu IS NULL)
   OR (p_adreça IS NULL)
   OR (p_municipi IS NULL)
   OR (p_telefon IS NULL)
 THEN
   RAISE campsNuls;
 END IF;
/*Es verifica que el document identificatiu no està duplicat */
 SELECT COUNT(*) INTO docIdDuplicat
 FROM Persona
 WHERE docIdentificatiu = p_docIdentificatiu;
 IF (docIdDuplicat > 0)
 THEN
   RAISE docIdentificatiuDuplicat;
 END IF;
/*Es verifica que el domicili triat existeix */
 SELECT COUNT(*) INTO noMunicipi
 FROM Municipi
 WHERE idMunicipi = p_municipi;
 IF (noMunicipi = 0)
 THEN
   RAISE noExisteixMunicipi;
 END IF;
/*S'insereix el nou registre a la taula Persona*/
 INSERT INTO Persona (nom, cognom1, cognom2, docIdentificatiu, adreça,
 municipi,  telefon, email)
   VALUES (p_nom, p_cognom1, p_cognom2, p_docIdentificatiu, p_adreça,
   p_municipi, p_telefon, p_email);
```

```
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'altaPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_ municipi || ', ' || p_telefon || ', ' || p_email, RSP);
COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'altaPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_ municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN docIdentificatiuDuplicat THEN
  RSP:= 'ERROR: Aquesta persona ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'altaPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_ municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN noExisteixMunicipi THEN
  RSP:= 'ERROR: El municipi indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'altaPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_ municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'altaPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_ municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta una persona';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'altaPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_ municipi || ', ' || p_telefon || ', ' || p_email, RSP);
END altaPersona;
```

(índex)

```
/*************************************************************************
Procediment que gestiona l'alta d'un alumne a la taula Alumne
Nom Procediment: altaAlumne
Paràmetres d'entrada: p_idPersona, p_numExpedient, p_dataNaixement,
p_anyIngres, p_nomPare, p_nomMare
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE altaAlumne (
  p_idPersona IN Number,
  p_numExpedient IN Varchar2,
  p_dataNaixement IN Date,
  p_anyIngres IN Char,
  p_nomPare IN Varchar2,
  p_nomMare IN Varchar2,
  RSP OUT Varchar2
)
IS
  expDuplicat Number;
  noPersona Number;
  noProfessor Number;
  nombreAlumnes Number;
  hihaNoAmonestats Number;
  nomAl Varchar2(60);
  p_any Varchar2(4);
  campsNuls EXCEPTION;
  expedientDuplicat EXCEPTION;
  noExisteixPersona EXCEPTION;
  esProfessor EXCEPTION;
/*Cursor per a obtenir els anys*/
  CURSOR c_anys IS
  SELECT distinct extract(year from data) as idAny
  FROM  Dates;
/*Tipus cursor*/
  cur_anys          c_anys%ROWTYPE;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_idPersona IS NULL)
    OR (p_numExpedient IS NULL)
    OR (p_dataNaixement IS NULL)
    OR (p_anyIngres IS NULL)
    OR (p_nomPare IS NULL)
    OR (p_nomMare IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que el número d'expedient no està duplicat */
  SELECT COUNT(*) INTO expDuplicat
  FROM Alumne
  WHERE numExpedient = p_numExpedient;
  IF (expDuplicat > 0)
  THEN
    RAISE expedientDuplicat;
  END IF;
/*Es verifica que s'ha donat d'alta el registre a la taula Persona*/
  SELECT COUNT(*) INTO noPersona
  FROM Persona
  WHERE idPersona = p_idPersona;
```

```
  IF (noPersona = 0)
  THEN
    RAISE noExisteixPersona;
  END IF;
/*Es verifica que la persona indicada no ha estat donada d'alta com a
 professor*/
  SELECT COUNT(*) INTO noProfessor
  FROM Professor
  WHERE idPersona = p_idPersona;
  IF (noProfessor > 0)
  THEN
    RAISE esProfessor;
  END IF;
/*S'insereix el nou registre a la taula Alumne */
  INSERT INTO Alumne (idPersona, numExpedient, dataNaixement, anyIngres,
  nomPare, nomMare)
    VALUES (p_idPersona, p_numExpedient, p_dataNaixement, p_anyIngres,
    p_nomPare, p_nomMare);
/*Actualització de la taula estadística del nombre d'amonestacions per
 alumne*/
  /*Es busca el nom de l'alumne i s'afegeix el nou registre a
    l'estadística, amb el nombre d'amonestacions a 0*/
  nomAl:= getNomAlumne(p_idPersona);
  INSERT INTO AmonestacionsAlumne (idAlumne, nomAlumne, nombreAmonAlumne)
    VALUES (p_idPersona, nomAl, 0);
/*Actualització de la taula estadística del nombre de sancions per
 alumne i any*/
  /*Es busquen els anys i s'afegeix el nou registre a l'estadística, amb
    el nombre de sancions a 0*/
  FOR cur_anys IN c_anys
    LOOP
      p_any:=cur_anys.idAny;
      INSERT    INTO    SancionsAlumne    (idAlumne,    nomAlumne,    idAny,
    nombreSancionsAlumne)
        VALUES (p_idPersona, nomAl, p_any, 0);
    END LOOP;
/*Actualització de la taula estadística dels alumnes que no tenen cap
    amonestació*/
  /*Es comprova si ja hi ha dades a la taula estadística*/
  SELECT COUNT(*) INTO hihaNoAmonestats
  FROM AlumnesNoAmonestats;
  /*Si no hi ha, s'hi insereix el primer registre*/
  IF (hihaNoAmonestats = 0) THEN
    INSERT INTO AlumnesNoAmonestats (nombreAlNoAmon) VALUES (1);
  ELSE
  /*Si n'hi havia, es modifica el nombre anterior afegint un més*/
    UPDATE AlumnesNoAmonestats
    SET nombreAlNoAmon = nombreAlNoAmon + 1;
  END IF;
  RSP:='OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', '||p_dataNaixement|| ', '||p_anyIngres|| ', '||p_nomPare|| ',
    '||p_nomMare, RSP);
COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
```

```
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', ' || p_dataNaixement || ', ' ||    p_anyIngres || ', ' ||
    p_nomPare || ', ' || p_nomMare, RSP);
  WHEN expedientDuplicat THEN
  RSP:= 'ERROR: Aquest número d'expedient ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', ' || p_dataNaixement || ', ' ||    p_anyIngres || ', ' ||
    p_nomPare || ', ' || p_nomMare, RSP);
  WHEN noExisteixPersona THEN
  RSP:= 'ERROR: Abans cal donar d'alta el registre a la taula Persona';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', ' || p_dataNaixement || ', ' ||    p_anyIngres || ', ' ||
    p_nomPare || ', ' || p_nomMare, RSP);
  WHEN esProfessor THEN
  RSP:= 'ERROR: Aquesta persona ha estat prèviament donada d'alta com a
        professor/a';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', ' || p_dataNaixement || ', ' ||    p_anyIngres || ', ' ||
    p_nomPare || ', ' || p_nomMare, RSP);
  WHEN dup_val_on_index THEN
  RSP:= 'ERROR: El número identificador de la persona està duplicat';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', ' || p_dataNaixement || ', ' ||    p_anyIngres || ', ' ||
    p_nomPare || ', ' || p_nomMare, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', ' || p_dataNaixement || ', ' ||    p_anyIngres || ', ' ||
    p_nomPare || ', ' || p_nomMare, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta un alumne';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAlumne', p_idPersona || ', ' || p_numExpedient
    || ', ' || p_dataNaixement || ', ' ||    p_anyIngres || ', ' ||
    p_nomPare || ', ' || p_nomMare, RSP);
END altaAlumne;
```

(índex)

```
/*************************************************************************
Procediment que gestiona l'alta d'un professor a la taula Professor
Nom Procediment: altaProfessor
Paràmetres d'entrada: p_idPersona, p_titulacio
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE altaProfessor (
  p_idPersona IN Number,
  p_titulacio IN Varchar2,
  RSP OUT Varchar2
)
IS
  noPersona Number;
  noAlumne Number;
  nomProfessor Varchar2(40);
  campsNuls EXCEPTION;
  noExisteixPersona EXCEPTION;
  esAlumne EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_idPersona IS NULL)
    OR (p_titulacio IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que s'ha donat d'alta el registre a Persona*/
  SELECT COUNT(*) INTO noPersona
  FROM Persona
  WHERE idPersona = p_idPersona;
  IF (noPersona = 0)
  THEN
    RAISE noExisteixPersona;
  END IF;
/*Es verifica que la persona indicada no ha estat donada d'alta com a
 alumne*/
  SELECT COUNT(*) INTO noAlumne
  FROM Alumne
  WHERE idPersona = p_idPersona;
  IF (noAlumne > 0)
  THEN
    RAISE esAlumne;
  END IF;
/*S'insereix el nou registre a la taula Professor*/
  INSERT INTO Professor (idPersona, titulacio)  VALUES (p_idPersona,
    p_titulacio);
/*Actualització de la taula estadística de la mitjana d'amonestacions
 per professor i any*/
   /*Es busca el nom del professor i s'afegeix el nou registre a
    l'estadística, amb la mitjana d'amonestacions a 0*/
  nomProfessor:= getNomProfessor(p_idPersona);
  INSERT  INTO  AvgAmonestacionsProfAny  (idProfessor,  nomProfessor,
    avgAmonProfAny)
  VALUES (p_idPersona, nomProfessor, 0);
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
```

```
    VALUES (SYSDATE, 'altaProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
  WHEN noExisteixPersona THEN
  RSP:= 'ERROR: Abans cal donar d'alta el registre a la taula Persona';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
  WHEN esAlumne THEN
  RSP:= 'ERROR: Aquesta persona ha estat prèviament donada d'alta com a
    alumne';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
  WHEN dup_val_on_index THEN
  RSP:= 'ERROR: El número identificador de la persona està duplicat';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta un professor';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
END altaProfessor;
```

(índex)

```
/*************************************************************************
Procediment que gestiona l'alta d'un curs a la taula Curs
Nom Procediment: altaCurs
Paràmetres d'entrada: p_nomCurs
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE altaCurs (
  p_nomCurs IN Varchar2,
  p_nombreGrups IN Number,
  RSP OUT Varchar2
)
IS
  noCurs Number;
  idC Number;
  p_any Varchar2(4);
  campsNuls EXCEPTION;
  cursDuplicat EXCEPTION;
  cursNoValid EXCEPTION;
/*Cursor per a obtenir els anys*/
  CURSOR c_anys IS
  SELECT distinct extract(year from data) as idAny
  FROM  Dates;
/*Tipus cursor*/
  cur_anys          c_anys%ROWTYPE;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_nomCurs IS NULL)
    OR (p_nombreGrups IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que el curs no està duplicat */
  SELECT COUNT(*) INTO noCurs
  FROM Curs
  WHERE nomCurs = p_nomCurs;
  IF (noCurs> 0)
  THEN
    RAISE cursDuplicat;
  END IF;
/*Es verifica que s'introdueix un valor correcte en el nom del curs*/
  IF (p_nomCurs NOT LIKE '1r. ESO' AND p_nomCurs NOT LIKE '2n. ESO' AND
    p_nomCurs NOT LIKE '3r. ESO' AND p_nomCurs NOT LIKE '4t. ESO' AND
    p_nomCurs NOT LIKE '1r. Batxillerat' AND p_nomCurs NOT LIKE '2n.
    Batxillerat')
  THEN
    RAISE cursNoValid;
  END IF;
/*S'insereix el nou registre a la taula Curs*/
  INSERT INTO Curs (nomCurs, p_nombreGrups) VALUES (p_nomCurs,
    p_nombreGrups);
/*Actualització de la taula estadística de la mitjana de sancions que
 tenen els alumnes per cursos*/
  /*Es busca l'identificador del curs i s'afegeix el nou registre a
    l'estadística, amb la mitjana de sancions a 0*/
  SELECT idCurs INTO idC
  FROM Curs
  WHERE nomCurs = p_nomCurs;
```

```
  INSERT INTO AvgSancionsCurs (idCurs, nomCurs, avgSancionsCurs)
    VALUES (idC, p_nomCurs, 0);
/*Actualització de la taula estadística del nombre de sancions per curs
 i any*/
  /*Es busquen els anys i s'afegeix el nou registre a l'estadística, amb
    el nombre de sancions a 0*/
  FOR cur_anys IN c_anys
    LOOP
      p_any:= cur_anys.idAny;
      INSERT    INTO    SancionsCursAny    (idCurs,    nomCurs,    idAny,
    nombreSancionsCurs)
        VALUES (idC, p_nomCurs, p_any, 0);
    END LOOP;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaCurs',  p_nomCurs  ||  ',  '  ||  p_nombreGrups,
    RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaCurs',  p_nomCurs  ||  ',  '  ||  p_nombreGrups,
    RSP);
  WHEN cursDuplicat THEN
  RSP:= 'ERROR: Aquest curs ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaCurs',  p_nomCurs  ||  ',  '  ||  p_nombreGrups,
    RSP);
  WHEN  cursNoValid THEN
  RSP:= 'ERROR: El nom del curs no és correcte';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaCurs',  p_nomCurs  ||  ',  '  ||  p_nombreGrups,
    RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaCurs',  p_nomCurs  ||  ',  '  ||  p_nombreGrups,
    RSP);
  WHEN  others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta un curs';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaCurs',  p_nomCurs  ||  ',  '  ||  p_nombreGrups,
    RSP);
END altaCurs;
```

(índex)

```
/*************************************************************************
Procediment que gestiona l'alta d'una assignatura a la taula Assignatura
Nom Procediment: altaAssignatura
Paràmetres d'entrada: p_idCurs, p_idAssignatura, p_nomAssignatura
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE altaAssignatura (
  p_idCurs IN Number,
  p_idAssignatura IN Number,
  p_nomAssignatura IN Varchar2,
  RSP OUT Varchar2
)
IS
  noCurs Number;
  noAssignatura Number;
  campsNuls EXCEPTION;
  noExisteixCurs EXCEPTION;
  assignaturaDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls*/
  IF (p_idCurs IS NULL)
    OR (p_idAssignatura IS NULL)
    OR (p_nomAssignatura IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que el curs a què s'ha d'assignar l'assignatura existeix*/
  SELECT COUNT(*) INTO noCurs
  FROM Curs
  WHERE idCurs = p_idCurs;
  IF (noCurs = 0)
  THEN
    RAISE noExisteixCurs;
  END IF;
/*Es verifica que l'assignatura no està duplicada */
  SELECT COUNT(*) INTO noAssignatura
  FROM Assignatura
  WHERE idCurs = p_idCurs AND idAssignatura = p_idAssignatura;
  IF (noAssignatura > 0)
  THEN
    RAISE assignaturaDuplicada;
  END IF;
/*S'insereix el nou registre a la taula Assignatura */
  INSERT INTO Assignatura (idCurs, idAssignatura, nomAssignatura)
    VALUES (p_idCurs, p_idAssignatura, p_nomAssignatura);
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
```

```
    VALUES   (SYSDATE,   'altaAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN noExisteixCurs THEN
  RSP:= 'ERROR: El curs indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN assignaturaDuplicada THEN
  RSP:= 'ERROR: Aquesta assignatura està duplicada';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta una assignatura';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
END altaAssignatura;
```

(índex)

```
/*************************************************************************
Procediment que gestiona l'alta d'una amonestació a la taula Amonestacio
Nom Procediment: altaAmonestacio
Paràmetres d'entrada: p_tipus, p_descripcio, p_gravetat
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE altaAmonestacio (
  p_tipus IN Varchar2,
  p_descripcio IN Varchar2,
  p_gravetat IN Varchar2,
  RSP OUT Varchar2
)
IS
  noAmonestacio NUMBER;
  campsNuls EXCEPTION;
  gravetatIncorrecta EXCEPTION;
  amonestacioDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_tipus IS NULL)
    OR (p_descripcio IS NULL)
    OR (p_gravetat IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que s'introdueixen valors correctes en el camp gravetat*/
  IF (p_gravetat NOT LIKE 'Lleu' AND p_gravetat NOT LIKE 'Greu' AND
    p_gravetat NOT LIKE 'Molt greu')
   THEN
    RAISE gravetatIncorrecta;
  END IF;
/*Es verifica que el tipus d'amonestació no està duplicat */
  SELECT COUNT(*) INTO noAmonestacio
  FROM Amonestacio
  WHERE tipus = p_tipus;
  IF (noAmonestacio> 0)
  THEN
    RAISE amonestacioDuplicada;
  END IF;
/*S'insereix el nou registre a la taula Amonestacio */
  INSERT INTO Amonestacio (tipus, descripcio, gravetat)
    VALUES (p_tipus,p_descripcio,p_gravetat);
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacio', p_tipus || ', ' || p_descripcio
    || ', ' || p_gravetat, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacio', p_tipus || ', ' || p_descripcio
    || ', ' || p_gravetat, RSP);
  WHEN gravetatIncorrecta THEN
```

```
  RSP:= 'ERROR: La gravetat de l''amonestació només pot ser Lleu, Greu o
        Molt greu';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacio', p_tipus || ', ' || p_descripcio
    || ', ' || p_gravetat, RSP);
  WHEN amonestacioDuplicada THEN
  RSP:= 'ERROR: El tipus d''amonestació ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacio', p_tipus || ', ' || p_descripcio
    || ', ' || p_gravetat, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacio', p_tipus || ', ' || p_descripcio
    || ', ' || p_gravetat, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta una amonestació';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacio', p_tipus || ', ' || p_descripcio
    || ', ' || p_gravetat, RSP);
END altaAmonestacio;
```

(índex)

```
/*********************************************************************
Procediment que gestiona l'alta d'una sanció a la taula Sancio
Nom Procediment: altaSancio
Paràmetres d'entrada: p_tipus, p_activacio
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE altaSancio (
  p_tipus IN Varchar2,
  p_activacio IN Varchar2,
  RSP OUT Varchar2
)
IS
  noSancio NUMBER;
  campsNuls EXCEPTION;
  sancioDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls*/
  IF (p_tipus IS NULL)
    OR (p_activacio IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que el tipus de sanció no està duplicat*/
  SELECT COUNT(*) INTO noSancio
  FROM Sancio
  WHERE tipus = p_tipus;
  IF (noSancio > 0) THEN
    RAISE sancioDuplicada;
  END IF;
/*S'insereix el nou registre a la taula Sancio*/
  INSERT INTO Sancio (tipus, activacio) VALUES (p_tipus, p_activacio);
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaSancio', p_tipus || ', ' || p_activacio, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaSancio', p_tipus || ', ' || p_activacio, RSP);
  WHEN sancioDuplicada THEN
  RSP:= 'ERROR: El tipus de sanció ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaSancio', p_tipus || ', ' || p_activacio, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaSancio', p_tipus || ', ' || p_activacio, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta una sanció';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaSancio', p_tipus || ', ' || p_activacio, RSP);
END altaSancio;
```

(índex)

```
/*************************************************************************
Procediment que gestiona l'alta d'un registre del calendari escolar a la
taula CalendariEscolar
Nom Procediment: altaCalendariEscolar
Paràmetres d'entrada: p_nomCalendari, p_data, p_curs, p_assignatura,
p_dia, p_hora
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/

CREATE OR REPLACE PROCEDURE altaCalendariEscolar (
  p_nomCalendari IN Varchar2,
  p_data IN Date,
  p_curs IN Number,
  p_assignatura IN Number,
  p_dia IN Number,
  p_hora IN Number,
  RSP OUT Varchar2
)
IS
  festiu Number;
  noAssignatura Number;
  noHora Number;
  noDuplicat Number;
  noData Number;
  campsNuls EXCEPTION;
  diaFestiu EXCEPTION;
  noExisteixAssignatura EXCEPTION;
  noExisteixHoraLectiva EXCEPTION;
  calendariDuplicat EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_nomCalendari IS NULL)
    OR (p_data IS NULL)
    OR (p_curs IS NULL)
    OR (p_assignatura IS NULL)
    OR (p_dia IS NULL)
    OR (p_hora IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que el dia indicat no és festiu*/
  SELECT COUNT(*) INTO festiu
  FROM Dates
  WHERE festiu = 'S' AND data = p_data;
  IF (festiu > 0)
  THEN
    RAISE diaFestiu;
  END IF;
/*Es verifica que s'ha donat d'alta l'assignatura*/
  SELECT COUNT(*) INTO noAssignatura
  FROM Assignatura
  WHERE idCurs = p_curs AND idAssignatura = p_assignatura;
  IF (noAssignatura = 0)
  THEN
    RAISE noExisteixAssignatura;
  END IF;
/*Es verifica que s'ha donat d'alta l'hora lectiva*/
```

```
  SELECT COUNT(*) INTO noHora
  FROM HoraLectiva
  WHERE idDia = p_dia AND horaDia = p_hora;
  IF (noHora= 0)
  THEN
    RAISE noExisteixHoraLectiva;
  END IF;
/*Es verifica que el registre no està duplicat */
  SELECT COUNT(*) INTO noDuplicat
  FROM CalendariEscolar
  WHERE data = p_data AND curs = p_curs AND assignatura = p_assignatura
    AND dia = p_dia AND hora = p_hora;
  IF (noDuplicat > 0)
  THEN
    RAISE calendariDuplicat;
  END IF;
/*Si la data indicada no hi és a la taula Dates, s'hi insereix*/
  SELECT COUNT(*) INTO noData
  FROM Dates
  WHERE data = p_data;
  IF (noData= 0)
  THEN
    INSERT INTO Dates (data, festiu) VALUES (p_data, 'N');
  END IF;
/*S'insereix el nou registre a la taula CalendariEscolar */
  INSERT INTO CalendariEscolar (nomCalendari ,data, curs, assignatura,
    dia, hora)
    VALUES (p_nomCalendari, p_data, p_curs, p_assignatura, p_dia,
    p_hora);
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' || p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' || p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
  WHEN diaFestiu THEN
  RSP:= 'ERROR: La data indicada correspon a un dia festiu';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' || p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
  WHEN noExisteixAssignatura THEN
  RSP:= 'ERROR: L''identificador del curs i/o de l''assignatura
        introduït no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' || p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
  WHEN noExisteixHoraLectiva THEN
  RSP:= 'ERROR: L''hora i/o dia indicat no existeix';
```

```
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' ||   p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
  WHEN calendariDuplicat THEN
  RSP:= 'ERROR: El registre de Calendari Escolar indicat ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' ||   p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' ||   p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta el calendari escolar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaCalendariEscolar', p_nomCalendari || ', ' ||
    p_data || ', ' || p_curs || ', ' ||   p_assignatura || ', ' || p_dia
    || ', ' || p_hora, RSP);
END altaCalendariEscolar;
```

(índex)

```
/*********************************************************************
Procediment que gestiona l'alta d'un registre a la taula
AmonestacionsEmeses. Es controla l'activació de les sancions
automàtiques i s'actualitzen estadístiques.
Nom Procediment: altaAmonestacionsEmeses
Paràmetres d'entrada: p_alumne, p_professor, p_amonestacio, p_curs,
p_assignatura, p_data, p_horaAmonestacio, p_comunicacioPares
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE altaAmonestacionsEmeses (
  p_alumne IN Number,
  p_professor IN Number,
  p_amonestacio IN Number,
  p_curs IN Number,
  p_assignatura IN Number,
  p_data IN DATE,
  p_horaAmonestacio IN CHAR,
  p_comunicacioPares IN CHAR,
  RSP OUT Varchar2
)
IS
  noAlumne Number;
  noProfessor Number;
  noAmonestacio Number;
  noCursAssignatura Number;
  noDuplicat Number;
  festiu Number;
  noData Number;
  acum Number;
  oper Char(1);
  reglaNombreAcum Number;
  sancioAut Number;
  amNomAlumne Varchar2(60);
  amNomProfessor Varchar2(60);
  amCurs Varchar2(20);
  amAny Char(4);
  noAmonestats Number;
  hihaAmonAlumne Number;
  nombreAmAlumne Number;
  hihaAvgProf Number;
  hihaProfMesAmon Number;
  profMesAmon Number;
  mitjana Number;
  campsNuls EXCEPTION;
  noExisteixAlumne EXCEPTION;
  noExisteixProfessor EXCEPTION;
  noExisteixAmonestacio EXCEPTION;
  noExisteixCursAssignatura EXCEPTION;
  diaFestiu EXCEPTION;
  amonEmesaDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_alumne IS NULL)
    OR (p_professor IS NULL)
    OR (p_amonestacio IS NULL)
    OR (p_curs IS NULL)
    OR (p_assignatura IS NULL)
```

```
    OR (p_data IS NULL)
    OR (p_horaAmonestacio IS NULL)
    OR (p_comunicacioPares IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que existeix l'alumne*/
  SELECT COUNT(*) INTO noAlumne
  FROM Alumne
  WHERE idPersona = p_alumne;
  IF (noAlumne = 0)
  THEN
    RAISE noExisteixAlumne;
  END IF;
/*Es verifica que existeix el professor*/
  SELECT COUNT(*) INTO noProfessor
  FROM Professor
  WHERE idPersona = p_professor;
  IF (noProfessor = 0)
  THEN
    RAISE noExisteixProfessor;
  END IF;
/*Es verifica que existeix l'amonestacio*/
  SELECT COUNT(*) INTO noAmonestacio
  FROM Amonestacio
  WHERE idAmonestacio = p_amonestacio;
  IF (noAmonestacio = 0)
  THEN
    RAISE noExisteixAmonestacio;
  END IF;
/*Es verifica que el professor imparteix l'assignatura especificada*/
  SELECT COUNT(*) INTO noCursAssignatura
  FROM Imparteix, Assignatura
  WHERE assignatura = idAssignatura AND curs = idCurs AND assignatura =
   p_assignatura AND curs = p_curs AND professor IN (SELECT  idPersona
                                                     FROM Professor
                                                     WHERE  professor  =
                                                     idPersona       AND
                                                     professor         =
                                                     p_professor);
  IF (noCursAssignatura = 0)
  THEN
    RAISE noExisteixCursAssignatura;
  END IF;
/*Es verifica que el dia indicat no és festiu*/
  SELECT COUNT(*)INTO festiu
  FROM Dates
  WHERE festiu = 'S' AND data = p_data;
  IF (festiu > 0)
  THEN
    RAISE diaFestiu;
  END IF;
/*Es verifica que el registre no està duplicat */
  SELECT COUNT(*) INTO noDuplicat
  FROM AmonestacionsEmeses
  WHERE alumne = p_alumne AND professor = p_professor AND amonestacio =
    p_amonestacio   AND   data   =   p_data   AND   horaAmonestacio   =
    p_horaAmonestacio;
  IF (noDuplicat > 0)
```

```
  THEN
    RAISE amonEmesaDuplicada;
  END IF;
/*S'insereix el nou registre a la taula AmonestacionsEmeses*/
  INSERT INTO AmonestacionsEmeses (alumne, professor, amonestacio, curs,
  assignatura, data, horaAmonestacio, comunicacioPares)
    VALUES (p_alumne, p_professor, p_amonestacio, p_curs, p_assignatura,
    p_data, p_horaAmonestacio, p_comunicacioPares);

/*Càlcul de si cal activar una sanció automàtica*/
  /*Es comprova si l'amonestació imposada correspon a alguna de les
  definides per activar automàticament una sanció*/
  SELECT COUNT(*) INTO corresponRegla
  FROM ReglaActivacio
  WHERE amonestacio = p_amonestacio;
  IF (corresponRegla > 0) THEN
    /*Es comprova si l'alumne tenia algun nombre acumulat per a aquest
    tipus d'amonestació*/
    SELECT COUNT (*) INTO acum
    FROM Acumulat
    WHERE idAlumne = p_alumne AND idAmonestacio = p_amonestacio;
    /*Si no hi havia, s'afegeix*/
    IF (acum = 0) THEN
       INSERT INTO Acumulat (idAlumne, idAmonestacio, nombreAcumulat)
             VALUES (p_alumne, p_amonestacio, 1);
    ELSE
    /*Si n'hi havia, s'afegeix una altra*/
    SELECT nombreAcumulat INTO acum
    FROM Acumulat
    WHERE idAlumne = p_alumne AND idAmonestacio = p_amonestacio;
    UPDATE Acumulat
    SET nombreAcumulat = acum + 1
    WHERE idAlumne = p_alumne AND idAmonestacio = p_amonestacio;
    END IF;
  /*S'obtenen les dades que defineixen la regla automàtica*/
    oper:= getOperadorRegla(p_amonestacio);
    reglaNombreAcum:= getNombreAcumulatRegla(p_amonestacio);
    sancioAut:= getSancioAutomatica(p_amonestacio);
    /*Si es donen les condicions definides a la regla, s'insereix una
    nova sanció, el nombre acumulat d'aquest tipus d'amonestació es posa
    a 0 i, en conseqüència, el registre desapareix de la taula Acumulat*/
    SELECT nombreAcumulat INTO acum
    FROM Acumulat
    WHERE idAlumne = p_alumne AND idAmonestacio = p_amonestacio;
    IF (oper = '=') AND (acum = reglaNombreAcum) THEN
       INSERT INTO SancionsEmeses (alumne, professor, sancio, data)
             VALUES (p_alumne, p_professor, sancioAut, p_data);
       DELETE FROM Acumulat
       WHERE idAlumne = p_alumne AND idAmonestacio = p_amonestacio ;
    /*I s'actualitzen les estadístiques*/
       estSancionsAlumne (p_alumne, p_data, RSP);
       estAlumneMesSancionat (p_alumne, p_data, RSP);
       estAvgSancionsCurs(p_alumne, RSP);
       estSancionsCursAny (p_alumne, p_data, RSP);
    END IF;
    IF (oper = '>') AND (acum > reglaNombreAcum) THEN
       INSERT INTO SancionsEmeses (alumne, professor, sancio, data)
             VALUES (p_alumne, p_professor, sancioAut, p_data);
       DELETE FROM Acumulat
```

```
      WHERE idAlumne = p_alumne AND idAmonestacio = p_amonestacio;
      estSancionsAlumne (p_alumne, p_data, RSP);
      estAlumneMesSancionat (p_alumne, p_data, RSP);
      estAvgSancionsCurs(p_alumne, RSP);
      estSancionsCursAny (p_alumne, p_data, RSP);
   END IF;
   IF (oper = '<') AND (acum < reglaNombreAcum) THEN
      INSERT INTO SancionsEmeses (alumne, professor, sancio, data)
            VALUES (p_alumne, p_professor, sancioAut, p_data);
      DELETE FROM Acumulat
      WHERE idAlumne = p_alumne AND idAmonestacio = p_amonestacio;
      estSancionsAlumne (p_alumne, p_data, RSP);
      estAlumneMesSancionat (p_alumne, p_data, RSP);
      estAvgSancionsCurs(p_alumne, RSP);
      estSancionsCursAny (p_alumne, p_data, RSP);
   END IF;
  END IF;
/*Actualització de l'estadística del nombre d'amonestacions per alumne*/
  estAmonestacionsAlumne(p_alumne, RSP);
/*Actualització de l'estadística del nombre d'alumnes que no tenen cap
  amonestació*/
  estAlumnesNoAmon(RSP);
/*Actualització de l'estadística de mitjana d'amonestacions dels
  professors per any*/
  estAvgAmonProfAny (p_professor, p_data, RSP);
/*Actualització de l'estadística del nom del professor més amonestador
  per curs*/
  estProfMesAmon(p_professor, p_alumne, p_curs, RSP);
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaAmonestacionsEmeses',  p_alumne  ||  ',  '  ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    ||  p_comunicacioPares, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaAmonestacionsEmeses',  p_alumne  ||  ',  '  ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    ||  p_comunicacioPares, RSP);
  WHEN noExisteixAlumne THEN
  RSP:= 'ERROR: L''identificador de l''alumne indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaAmonestacionsEmeses',  p_alumne  ||  ',  '  ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    ||  p_comunicacioPares, RSP);
  WHEN noExisteixProfessor THEN
  RSP:= 'ERROR: L''identificador del professor indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaAmonestacionsEmeses',  p_alumne  ||  ',  '  ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    ||  p_comunicacioPares, RSP);
```

```
  WHEN noExisteixAmonestacio THEN
  RSP:= 'ERROR: L''identificador indicat de l''amonestacio no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacionsEmeses', p_alumne || ', ' ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    || p_comunicacioPares, RSP);
  WHEN noExisteixCursAssignatura THEN
    RSP:= 'ERROR: El professor indicat no imparteix l''assignatura
    assenyalada';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacionsEmeses', p_alumne || ', ' ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    || p_comunicacioPares, RSP);
  WHEN diaFestiu THEN
  RSP:= 'ERROR: La data indicada correspon a un dia festiu';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacionsEmeses', p_alumne || ', ' ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    || p_comunicacioPares, RSP);
  WHEN amonEmesaDuplicada THEN
  RSP:= 'ERROR: El registre indicat d''AmonestacionsEmeses ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacionsEmeses', p_alumne || ', ' ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    || p_comunicacioPares, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacionsEmeses', p_alumne || ', ' ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    || p_comunicacioPares, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar d''alta el registre a les
    amonestacions emeses';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'altaAmonestacionsEmeses', p_alumne || ', ' ||
    p_professor || ', ' || p_amonestacio || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_data || ', ' || p_horaAmonestacio || ', '
    || p_comunicacioPares, RSP);
END altaAmonestacionsEmeses;
```

(índex)

```
/*********************************************************************
Procediment que gestiona l'alta d'un registre a la taula SancionsEmeses.
També es controla l'actualització d'estadístiques.
Nom Procediment: altaSancionsEmeses
Paràmetres d'entrada: p_alumne, p_professor, p_sancio, p_data
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
   d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE altaSancionsEmeses (
  p_alumne IN Number,
  p_professor IN Number,
  p_sancio IN Number,
  p_data IN DATE,
  RSP OUT Varchar2
)
IS
  noAlumne Number;
  noProfessor Number;
  noSancio Number;
  festiu Number;
  noDuplicat Number;
  noData Number;
  sancioNomAlumne Varchar2(60);
  sancioNomProfessor Varchar2(60);
  sancioNomCurs Varchar2(20);
  sIdCurs Number;
  sancioAny Char(4);
  hihaSancioAlumne Number;
  hihaSancioCurs Number;
  hihaAvgCurs Number;
  hihaAlumneMesSan Number;
  nombreSanAlumne Number;
  nombreSanCurs Number;
  alumneMesSan Number;
  mitjana Number;
  campsNuls EXCEPTION;
  noExisteixAlumne EXCEPTION;
  noExisteixProfessor EXCEPTION;
  noExisteixSancio EXCEPTION;
  diaFestiu EXCEPTION;
  sancioEmesaDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls*/
  IF (p_alumne IS NULL)
    OR (p_professor IS NULL)
    OR (p_sancio IS NULL)
    OR (p_data IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que existeix l'alumne*/
  SELECT COUNT(*) INTO noAlumne
  FROM Alumne
  WHERE idPersona = p_alumne;
  IF (noAlumne = 0)
  THEN
    RAISE noExisteixAlumne;
  END IF;
```

```
/*Es verifica que existeix el professor*/
  SELECT COUNT(*) INTO noProfessor
  FROM Professor
  WHERE idPersona = p_professor;
  IF (noProfessor = 0)
  THEN
    RAISE noExisteixProfessor;
  END IF;
/*Es verifica que existeix la sanció*/
  SELECT COUNT(*) INTO noSancio
  FROM Sancio
  WHERE idSancio = p_sancio;
  IF (noSancio = 0)
  THEN
    RAISE noExisteixSancio;
  END IF;
/*Es verifica que el dia indicat no és festiu*/
  SELECT COUNT(*) INTO festiu
  FROM Dates
  WHERE festiu = 'S' AND data = p_data;
  IF (festiu > 0)
  THEN
    RAISE diaFestiu;
  END IF;
/*Es verifica que el registre no està duplicat */
  SELECT COUNT(*) INTO noDuplicat
  FROM SancionsEmeses
  WHERE alumne = p_alumne AND professor = p_professor AND sancio =
    p_sancio AND data = p_data;
  IF (noDuplicat > 0)
  THEN
    RAISE sancioEmesaDuplicada;
  END IF;
/*S'insereix el nou registre a la taula SancionsEmeses*/
 INSERT INTO SancionsEmeses (alumne, professor, sancio, data)
    VALUES (p_alumne, p_professor, p_sancio, p_data);
/*Actualització de l'estadística del nombre de sancions per alumne i
  any*/
  estSancionsAlumne (p_alumne, p_data, RSP);
/*Actualització de l'estadística del nombre de sancions per curs i any*/
  estSancionsCursAny (p_alumne, p_data, RSP);
/*Actualització de l'estadística de la mitjana de sancions que tenen els
  alumnes per curs*/
 estAvgSancionsCurs (p_alumne, RSP);
/*Actualització de l'estadística del nom de l'alumne més sancionat en un
  any donat*/
  estAlumneMesSancionat (p_alumne, p_data, RSP);
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'altaSancionsEmeses',  p_alumne  ||  ',  '  ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
```

```
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  WHEN noExisteixAlumne THEN
  RSP:= 'ERROR: L''identificador de l''alumne indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  WHEN noExisteixProfessor THEN
  RSP:= 'ERROR: L''identificador del professor indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  WHEN noExisteixSancio THEN
  RSP:= 'ERROR: L''identificador indicat de la sanció no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  WHEN diaFestiu THEN
  RSP:= 'ERROR: La data indicada correspon a un dia festiu';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  WHEN sancioEmesaDuplicada THEN
  RSP:= 'ERROR: El registre indicat ja existeix a SancionsEmeses';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
  WHEN others THEN
  RSP:=  'ERROR:  Error  genèric  en  donar  d''alta  el  registre  a  les
    sancions emeses';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'altaSancionsEmeses',   p_alumne   ||   ',   '   ||
    p_professor || ', ' || p_sancio || ', ' || p_data, RSP);
END altaSancionsEmeses;
```

(índex)

```
/*********************************************************************
Procediment que gestiona la baixa d'una persona a la taula Persona a
partir del seu document identificatiu. El procediment és el mateix tant
si es tracta de gestionar la baixa d'un alumne com d'un professor.
S'eliminen, també, els registres d'aquelles taules en què la persona
estigui referenciada, però no s'elimina de la taula que contingui les
dades històriques. A les taules estadístiques que corresponguin, s'hi
insereix una observació indicativa de la baixa.
Nom Procediment: baixaPersona
Paràmetres d'entrada: p_docIdentificatiu
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE baixaPersona (
  p_docIdentificatiu IN Varchar2,
  RSP OUT Varchar2
)
IS
  tipus Number;
  noDocIdentificatiu Number;
  noExisteixDocIdentificatiu EXCEPTION;
BEGIN
/*Es verifica que el document identificatiu introduït existeix*/
  SELECT COUNT(*) INTO noDocIdentificatiu
  FROM Persona
  WHERE docIdentificatiu = p_docIdentificatiu;
  IF (noDocIdentificatiu = 0)
  THEN
    RAISE noExisteixDocIdentificatiu;
  END IF;
/*Es comprova si és alumne o professor i s'elimina el registre de totes
les taules en què n'hi hagi de relacionats*/
  SELECT COUNT(*) INTO tipus
  FROM Alumne a, Persona b
  WHERE    a.idPersona    =    b.idPersona    AND    b.docIdentificatiu    =
          p_docIdentificatiu;
  IF (tipus = 0) THEN
  DELETE FROM AmonestacionsEmeses
  WHERE professor IN (SELECT r.idPersona
                      FROM Persona r
                      WHERE r.docIdentificatiu = p_docIdentificatiu);
  DELETE FROM SancionsEmeses
  WHERE professor IN (SELECT r.idPersona
                      FROM Persona r
                      WHERE r.docIdentificatiu = p_docIdentificatiu);
  DELETE FROM Es_tutor
  WHERE professor IN (SELECT r.idPersona
                      FROM Persona r
                      WHERE r.docIdentificatiu = p_docIdentificatiu);
  DELETE FROM Imparteix
  WHERE professor IN (SELECT r.idPersona
                      FROM Persona r
                      WHERE r.docIdentificatiu = p_docIdentificatiu);
  DELETE FROM Professor
  WHERE idPersona IN (SELECT r.idPersona
                      FROM Persona r
                      WHERE r.docIdentificatiu = p_docIdentificatiu);
```

```
/*S'indica a les taules d'amonestacions dels professors que aquest en
 concret s'ha donat de baixa, però no s'elimina el registre per tal que
 es mantinguin les dades estadístiques*/
 UPDATE AvgAmonestacionsProfAny
 SET observacions = 'Professor/a que s'ha donat de baixa'
 WHERE idProfessor IN (SELECT r.idPersona
                       FROM Persona r
                       WHERE r.docIdentificatiu = p_docIdentificatiu);
 UPDATE ProfessorMesAmonestador
 SET observacions = 'Professor/a que s'ha donat de baixa'
 WHERE idProfessor IN (SELECT r.idPersona
                       FROM Persona r
                       WHERE r.docIdentificatiu = p_docIdentificatiu);
 END IF;
 IF (tipus > 0) THEN
 DELETE FROM Matriculat
 WHERE alumne IN (SELECT r.idPersona
                  FROM Persona r
                  WHERE r.docIdentificatiu = p_docIdentificatiu);
 DELETE FROM AmonestacionsEmeses
 WHERE alumne IN (SELECTr.idPersona
                  FROM Persona r
                  WHERE r.docIdentificatiu = p_docIdentificatiu);
 DELETE FROM SancionsEmeses
 WHERE alumne IN(SELECT r.idPersona
                  FROM Persona r
                  WHERE r.docIdentificatiu = p_docIdentificatiu);
 DELETE FROM Acumulat
 WHERE idAlumne IN(SELECT r.idPersona
                   FROM Persona r
                   WHERE r.docIdentificatiu = p_docIdentificatiu);
 DELETE FROM Alumne
 WHERE idPersona IN(SELECT r.idPersona
                    FROM Persona r
                    WHERE r.docIdentificatiu = p_docIdentificatiu);
/*S'indica a les taules d'amonestacions i sancions dels alumnes que
 aquest en concret s'ha donat de baixa, però no s'elimina el registre
 per tal que es mantinguin les dades estadístiques*/
 UPDATE AmonestacionsAlumne
 SET    observacions = 'Alumne donat de baixa'
 WHERE idAlumne IN (SELECT r.idPersona
                    FROM Persona r
                    WHERE r.docIdentificatiu = p_docIdentificatiu);
 UPDATE AlumneMesSancionat
 SET    observacions = 'Alumne donat de baixa'
 WHERE idAlumne IN (SELECT r.idPersona
                    FROM Persona r
                    WHERE r.docIdentificatiu = p_docIdentificatiu);
 UPDATE SancionsAlumne
 SET    observacions = 'Alumne donat de baixa'
 WHERE idAlumne IN (SELECT r.idPersona
                    FROM Persona r
                    WHERE r.docIdentificatiu = p_docIdentificatiu);
 END IF;
/*S'elimina el registre de la taula Persona */
 DELETE FROM Persona
 WHERE docIdentificatiu = p_docIdentificatiu;
 RSP:= 'OK';
/*S'actualitza la taula log */
```

```
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaPersona', p_docIdentificatiu, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN noExisteixDocIdentificatiu THEN
  RSP:=  'ERROR:  No  s''ha  trobat  cap  persona  amb  el  document
          identificatiu indicat';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaPersona', p_docIdentificatiu, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut donar de baixa';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaPersona', p_docIdentificatiu, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar de baixa la persona';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaPersona', p_docIdentificatiu, RSP);
END baixaPersona;
```

(índex)

```
/*********************************************************************
Procediment que gestiona la baixa d'un curs a la taula Curs a partir del
seu identificador.La baixa del curs comporta, també, la baixa de les
assignatures que té assignades; a més, s'eliminenels registres
d'aquelles taules en què el curs estigui referenciat, però no s'elimina
de la taulaque conté les dades històriques. A les taules estadístiques
que corresponguin, s'hi insereix una observació indicativa de la baixa.
Nom Procediment: baixaCurs
Paràmetres d'entrada: p_idCurs
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE baixaCurs (
  p_idCurs IN Number,
  RSP OUT Varchar2
)
IS
  noCurs Number;
  noExisteixCurs EXCEPTION;
BEGIN
/*Es verifica que el curs existeixi*/
  SELECT COUNT(*) INTO noCurs
  FROM Curs
  WHERE idCurs = p_idCurs;
  IF (noCurs = 0)
  THEN
    RAISE noExisteixCurs;
  END IF;
/*S'eliminen els registres en aquelles taules en què el curs s'hi
 relaciona*/
  DELETE FROM AmonestacionsEmeses
  WHERE curs = p_idCurs;
  DELETE FROM Es_tutor
  WHERE curs = p_idCurs;
  DELETE FROM Matriculat
  WHERE curs = p_idCurs;
  DELETE FROM Imparteix
  WHERE curs = p_idCurs;
  DELETE FROM CalendariEscolar
  WHERE curs = p_idCurs;
  DELETE FROM Assignatura
  WHERE idCurs = p_idCurs;
/*S'indica a les taules estadístiques relatives al cursos que aquest en
 concret s'ha donat de baixa, però no s'elimina el registre per tal que
 es mantinguin les dades estadístiques*/
  UPDATE SancionsCursAny
  SET    observacions = 'Curs donat de baixa'
  WHERE idCurs IN (SELECT c.idCurs
                   FROM Curs c
                   WHERE c.idCurs = p_idCurs);
  UPDATE AvgSancionsCurs
  SET    observacions = 'Curs donat de baixa'
  WHERE idCursIN (SELECT c.idCurs
                   FROM Curs c
                   WHERE c.idCurs = p_idCurs);
/*S'elimina el registre de la taula Curs */
  DELETE FROM Curs
  WHERE idCurs = p_idCurs;
```

```
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaCurs', p_idCurs, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN noExisteixCurs THEN
  RSP:= 'ERROR: No s''ha trobat cap curs amb l''identificador indicat';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaCurs', p_idCurs, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut donar de baixa';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaCurs', p_idCurs, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar de baixa el curs';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaCurs', p_idCurs, RSP);
END baixaCurs;
```

(índex)

```
/*********************************************************************
Procediment que gestiona la baixa d'una assignatura a la taula
Assignatura a partir dels seus identificadors. S'eliminen, també, els
registres d'aquelles taules en què l'assignatura estigui referenciada,
però no s'elimina de la taula que contingui les dades històriques.
Nom Procediment: baixaAssignatura
Paràmetres d'entrada: p_idCurs, p_idAssignatura
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE baixaAssignatura (
  p_idCurs IN Number,
  p_idAssignatura IN Number,
  RSP OUT Varchar2
)
IS
  noAssignatura Number;
  campsNuls EXCEPTION;
  noExisteixAssignatura EXCEPTION;
BEGIN
/*Es verifica que els identificadors de l'assignatura no siguin nuls*/
  IF (p_idCurs IS NULL)
    OR (p_idAssignatura IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que l'assignatura existeixi*/
  SELECT COUNT(*) INTO noAssignatura
  FROM Assignatura
  WHERE idCurs = p_idCurs AND idAssignatura = p_idAssignatura;
  IF (noAssignatura = 0)
  THEN
    RAISE noExisteixAssignatura;
  END IF;
/*S'eliminen els registres en aquelles taules en què l'assignatura s'hi
 relaciona*/
  DELETE FROM Imparteix
  WHERE curs = p_idCurs AND assignatura = p_idAssignatura;
  DELETE FROM CalendariEscolar
  WHERE curs = p_idCurs AND assignatura = p_idAssignatura;
  DELETE FROM AmonestacionsEmeses
  WHERE curs = p_idCurs AND assignatura = p_idAssignatura;
/*S'elimina el registre de la taula Assignatura */
  DELETE FROM Assignatura
  WHERE idCurs = p_idCurs AND idAssignatura = p_idAssignatura;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES    (SYSDATE,'baixaAssignatura',p_idCurs    ||    ',    '    ||
    p_idAssignatura, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Els identificadors de l''assignatura no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES    (SYSDATE,'baixaAssignatura',p_idCurs    ||    ',    '    ||
    p_idAssignatura, RSP);
```

```
  WHEN noExisteixAssignatura THEN
  RSP:= 'ERROR: No s''ha trobat cap assignatura amb els identificadors
    indicats';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES    (SYSDATE,'baixaAssignatura',p_idCurs    ||    ',    '    ||
    p_idAssignatura, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut donar de baixa';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES    (SYSDATE,'baixaAssignatura',p_idCurs    ||    ',    '    ||
    p_idAssignatura, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar de baixa l''assignatura';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES    (SYSDATE,'baixaAssignatura',p_idCurs    ||    ',    '    ||
    p_idAssignatura, RSP);
END baixaAssignatura;
```

(índex)

```
/*************************************************************************
Procediment que gestiona la baixa d'un tipus d'amonestació a la
taulaAmonestacio a partir del seu identificador. S'eliminen, també, els
registres d'aquelles taules en què l'amonestació estigui referenciada,
però no s'elimina de la taula que contingui les dades històriques.
Nom Procediment: baixaAmonestacio
Paràmetres d'entrada: p_idAmonestacio
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE baixaAmonestacio (
  p_idAmonestacio IN Number,
  RSP OUT Varchar2
)
IS
  noAmonestacio NUMBER;
  noExisteixAmonestacio EXCEPTION;
BEGIN
/*Es verifica que l'amonestacio existeixi */
  SELECT COUNT(*) INTO noAmonestacio
  FROM Amonestacio
  WHERE idAmonestacio = p_idAmonestacio;
  IF (noAmonestacio = 0) THEN
    RAISE noExisteixAmonestacio;
  END IF;
/*S'eliminen els registres en aquelles taules amb què s'hi relaciona*/
  DELETE FROM AmonestacionsEmeses
  WHERE amonestacio = p_idAmonestacio;
  DELETE FROM Acumulat
  WHERE idAmonestacio = p_idAmonestacio;
  DELETE FROM ReglaActivacio
  WHERE amonestacio = p_idAmonestacio;
/*S'elimina el tipus d'amonestació de la taula Amonestacio */
  DELETE FROM Amonestacio
  WHERE idAmonestacio = p_idAmonestacio;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaAmonestacio',p_idAmonestacio, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN noExisteixAmonestacio THEN
  RSP:= 'ERROR: No s''ha trobat cap amonestacio amb l''identificador
        indicat';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaAmonestacio',p_idAmonestacio, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut donar de baixa';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaAmonestacio',p_idAmonestacio, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar de baixa l''amonestació';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaAmonestacio',p_idAmonestacio, RSP);
END baixaAmonestacio;
```

(índex)

```
/*************************************************************************
Procediment que gestiona la baixa d'un tipus de sanció a la taula Sancio
a partir del seu identificador. S'eliminen, també, els registres
d'aquelles taules en què la sanció estigui referenciada, però no
s'elimina de la taulaque conté les dades històriques.
Nom Procediment: baixaSancio
Paràmetres d'entrada: p_idSancio
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE baixaSancio (
  p_idSancio IN Number,
  RSP OUT Varchar2
)
IS
  noSancio Number;
  noExisteixSancio EXCEPTION;
BEGIN
/*Es verifica que la sanció existeixi */
  SELECT COUNT(*) INTO noSancio
  FROM Sancio
  WHERE idSancio = p_idSancio;
  IF (noSancio = 0)
  THEN
    RAISE noExisteixSancio;
  END IF;
/*S'eliminen els registres en aquelles taules en què la sanció s'hi
 relaciona*/
  DELETE FROM SancionsEmeses
  WHERE sancio = p_idSancio;
  DELETE FROM ReglaActivacio
  WHERE sancio = p_idSancio;
/*S'elimina el registre de la taula Sancio*/
  DELETE FROM Sancio
  WHERE idSancio = p_idSancio;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaSancio',p_idSancio, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN noExisteixSancio THEN
  RSP:=  'ERROR:  No  s''ha  trobat  cap  sanció  amb  l''identificador
          indicat';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaSancio', p_idSancio, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut donar de baixa';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaSancio', p_idSancio, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar de baixa la sanció';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'baixaSancio', p_idSancio, RSP);
END baixaSancio;
```

(índex)

```
/*********************************************************************
Procediment que gestiona la baixa d'un registre d'un calendari escolar a
la taulaCalendariEscolar a partir del seu identificador. No s'elimina de
la taula que contingui les dades històriques.
Nom Procediment: baixaCalendariEscolar
Paràmetres d'entrada: p_idCalendari
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE baixaCalendariEscolar (
  p_idCalendari IN Number,
  RSP OUT Varchar2
)
IS
  noCalendari Number;
  noExisteixRegistreCalendari EXCEPTION;
BEGIN
/*Es verifica que el registre que es vol donar de baixa existeixi */
  SELECT COUNT(*) INTO noCalendari
  FROM CalendariEscolar
  WHERE idCalendari = p_idCalendari;
  IF (noCalendari = 0)
  THEN
    RAISE noExisteixRegistreCalendari;
  END IF;
/*S'elimina el registre de la taulaCalendariEscolar */
  DELETE FROM CalendariEscolar
  WHERE idCalendari = p_idCalendari;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'baixaCalendariEscolar', p_idCalendari, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN noExisteixRegistreCalendari THEN
  RSP:= 'ERROR: No s''ha trobat cap registre en Calendari Escolar amb
        l''identificador indicat';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'baixaCalendariEscolar', p_idCalendari, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut donar de baixa';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'baixaCalendariEscolar', p_idCalendari, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en donar de baixa el registre del
         Calendari Escolar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'baixaCalendariEscolar', p_idCalendari, RSP);
END baixaCalendariEscolar;
```

(índex)

```
/*************************************************************************
Procediment que gestiona la modificació de les dades d'una persona a la
taula Persona a partir del seu document identificatiu, que no pot canviar
Nom Procediment: modifPersona
Paràmetres d'entrada: p_nom, p_cognom1, p_cognom2, p_docIdentificatiu,
p_adreça, p_codiPostal, p_telefon, p_email
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE modifPersona (
  p_nom IN Varchar2,
  p_cognom1 IN Varchar2,
  p_cognom2 IN Varchar2,
  p_docIdentificatiu IN Varchar2,
  p_adreça IN Varchar2,
  p_municipi IN Number,
  p_telefon IN Number,
  p_email IN Varchar2,
  RSP OUT Varchar2
)
IS
  noDocIdentificatiu Number;
  noMunicipi Number;
  idC Number;
  noIdC Number;
  tipus Number;
  nom Varchar2(40);
  campsNuls EXCEPTION;
  noExisteixPersona EXCEPTION;
  noExisteixMunicipi EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF(p_nom IS NULL)
    OR (p_cognom1 IS NULL)
    OR (p_docIdentificatiu IS NULL)
    OR (p_adreça IS NULL)
    OR (p_Municipi IS NULL)
    OR (p_telefon IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que la persona indicada existeixi */
  SELECT COUNT(*) INTO noDocIdentificatiu
  FROM Persona
  WHERE docIdentificatiu = p_docIdentificatiu;
  IF (noDocIdentificatiu = 0) THEN
    RAISE noExisteixPersona;
  END IF;
/*Es verifica que el domicili que es modifiqui existeixi */
  SELECT COUNT(*) INTO noMunicipi
  FROM Municipi
  WHERE idMunicipi = p_municipi;
  IF (noMunicipi = 0) THEN
    RAISE noExisteixMunicipi;
  END IF;
/*Es modifica el registre de la taula Persona */
  UPDATE Persona
  SET  nom = p_nom,
```

```
      cognom1 = p_cognom1,
      cognom2 = p_cognom2,
      adreça = p_adreça,
      municipi = p_municipi,
      telefon = p_telefon,
      email = p_email
  WHERE docIdentificatiu = p_docIdentificatiu;
/*Es comprova si la persona que es modifica és un alumne o un professor
  per tal d'actualitzar l'estadística corresponent*/
  SELECT COUNT(*) INTO tipus
  FROM Alumne a, Persona b
  WHERE    a.idPersona    =    b.idPersona    AND    b.docIdentificatiu    =
          p_docIdentificatiu;
  IF (tipus > 0) THEN
  /*S'actualitza la taula estadística del nombre d'amonestacions per
  alumne si ha hagut modificació en el nom d'un alumne*/
    SELECT distinct a.idAlumne INTO idC
    FROM AmonestacionsAlumne a, Persona r
    WHERE    a.idAlumne    =    r.idPersona    AND    r.docIdentificatiu    =
    p_docIdentificatiu;
    nom:= getNomAlumne(idC);
    UPDATE AmonestacionsAlumne
    SET nomAlumne = nom
    WHERE idAlumne = idC;
  /*S'actualitza la taula de sancions dels alumnes si ha hagut
  modificació en el nom de l'alumne*/
    SELECT distinct s.idAlumne INTO idC
    FROM SancionsAlumne s, Persona r
    WHERE    s.idAlumne    =    r.idPersona    AND    r.docIdentificatiu    =
    p_docIdentificatiu;
    UPDATE SancionsAlumne
    SET nomAlumne = nom
    WHERE idAlumne = idC;
  /*Si és el cas, s'actualitza la taula de l'alumne més sancionat per
  anys si ha hagut modificació en el nom de l'alumne*/
    SELECT COUNT(*) INTO noIdC
    FROM AlumneMesSancionat, Persona
    WHERE idAlumne = idPersona AND docIdentificatiu = p_docIdentificatiu;
    IF (noIdC > 0) THEN
       SELECT distinct s.idAlumne INTO idC
       FROM AlumneMesSancionat s, Persona r
       WHERE    s.idAlumne    =    r.idPersona    AND    r.docIdentificatiu    =
              p_docIdentificatiu;
       UPDATE AlumneMesSancionat
       SET   nomAlumne = nom
       WHERE idAlumne = idC;
    END IF;
  ELSE
  /*Si és el cas, s'actualitza la taula del professor més amonestador
    per curs si ha hagut modificació en el nom del professor*/
    SELECT COUNT(*) INTO noIdC
    FROM ProfessorMesAmonestador, Persona
     WHERE    idProfessor    =    idPersona    AND    docIdentificatiu    =
            p_docIdentificatiu;
     IF (noIdC > 0) THEN
          SELECT s.idProfessor INTO idC
          FROM ProfessorMesAmonestador s, Persona r
          WHERE  s.idProfessor  =  r.idPersona  AND  r.docIdentificatiu  =
                 p_docIdentificatiu;
```

```
          nom:= getNomProfessor(idC);
          UPDATE ProfessorMesAmonestador
          SET    nomProfessor = nom
          WHERE idProfessor = idC;
    END IF;
  /*S'actualitza la taula de mitjana d'amonestacions per professor i any
    si ha hagut modificació en el nom del professor*/
    SELECT s.idProfessor INTO idC
    FROM AvgAmonestacionsProfAny s, Persona r
    WHERE   s.idProfessor   =   r.idPersona   AND   r.docIdentificatiu   =
    p_docIdentificatiu;
    UPDATE AvgAmonestacionsProfAny
    SET nomProfessor = nom
    WHERE idProfessor = idC;
  END IF;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'modifPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'modifPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN noExisteixMunicipi THEN
  RSP:= 'ERROR: El municipi indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'modifPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN noExisteixPersona THEN
  RSP:= 'ERROR: Es fa referència a una persona que no existeix a la base
  de dades o ha estat donada de baixa';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'modifPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut inserir';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'modifPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_municipi || ', ' || p_telefon || ', ' || p_email, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en modificar les dades  d''una persona';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE,'modifPersona', p_nom || ', ' || p_cognom1 || ', ' ||
    p_cognom2 || ', ' ||   p_docIdentificatiu || ', ' || p_adreça || ', '
    || p_municipi || ', ' || p_telefon || ', ' || p_email, RSP);
END modifPersona;
```

(índex)

```
/*****************************************************************
Procediment que gestiona la modificació de les dades d'un alumne a la
taula Alumne a partir de l'identificador de la persona a què es fa
referència. No es permet modificar el número d'expedient escolar
Nom Procediment: modifAlumne
Paràmetres d'entrada: p_idPersona, p_numExpedient, p_dataNaixement,
p_anyIngres, p_nomPare, p_nomMare
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*****************************************************************/
CREATE OR REPLACE PROCEDURE modifAlumne (
  p_idPersona IN Number,
  p_dataNaixement IN Date,
  p_anyIngres IN Char,
  p_nomPare IN Varchar2,
  p_nomMare IN Varchar2,
  RSP OUT Varchar2
)
IS
  noAlumne Number;
  campsNuls EXCEPTION;
  noExisteixAlumne EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_idPersona IS NULL)
    OR (p_dataNaixement IS NULL)
    OR (p_anyIngres IS NULL)
    OR (p_nomPare IS NULL)
    OR (p_nomMare IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que existeixi l'alumne a què fa referència
 l'identificador*/
  SELECT COUNT(*) INTO noAlumne
  FROM Alumne
  WHERE idPersona = p_idPersona;
  IF (noAlumne = 0)
  THEN
    RAISE noExisteixAlumne;
  END IF;
/*Es modifica el registre a la taula Alumne*/
  UPDATE Alumne
  SET  dataNaixement = p_dataNaixement,
       anyIngres = p_anyIngres,
       nomPare = p_nomPare,
       nomMare = p_nomMare
  WHERE idPersona = p_idPersona;
  RSP:='OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES    (SYSDATE,    'modifAlumne',    p_idPersona    ||    ',    '||
    p_dataNaixement||    ',    '||p_anyIngres||    ',    '||p_nomPare||    ',
    '||p_nomMare, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
```

```
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAlumne',   p_idPersona   ||   ',   '   ||
    p_dataNaixement || ', ' ||   p_anyIngres || ', ' || p_nomPare || ', '
    || p_nomMare, RSP);
  WHEN noExisteixAlumne THEN
  RSP:= 'ERROR: L'identificador introduït no correspon a cap alumne';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAlumne',   p_idPersona   ||   ',   '   ||
    p_dataNaixement || ', ' ||   p_anyIngres || ', ' || p_nomPare || ', '
    || p_nomMare, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut modificar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAlumne',   p_idPersona   ||   ',   '   ||
    p_dataNaixement || ', ' ||   p_anyIngres || ', ' || p_nomPare || ', '
    || p_nomMare, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en modificar les dades de l''alumne';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAlumne',   p_idPersona   ||   ',   '   ||
    p_dataNaixement || ', ' ||   p_anyIngres || ', ' || p_nomPare || ', '
    || p_nomMare, RSP);
END modifAlumne;
```

(índex)

```
/*************************************************************************
Procediment que gestiona la modificació de les dades d'un professor a la
taula Professora partir de l'identificador de la persona a què es fa
referència.
Nom Procediment: modifProfessor
Paràmetres d'entrada: p_idPersona, p_titulacio
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE modifProfessor (
  p_idPersona IN Number,
  p_titulacio IN Varchar2,
  RSP OUT Varchar2
)
IS
  noProfessor Number;
  campsNuls EXCEPTION;
  noExisteixProfessor EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_idPersona IS NULL)
    OR (p_titulacio IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que existeixi el professor a què fa referència
 l'identificador*/
  SELECT COUNT(*) INTO noProfessor
  FROM Professor
  WHERE idPersona = p_idPersona;
  IF (noProfessor = 0)
  THEN
    RAISE noExisteixProfessor;
  END IF;
/*Es modifica el registre a la taula Professor */
  UPDATE Professor
  SET  titulacio = p_titulacio
  WHERE idPersona = p_idPersona;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifProfessor',   p_idPersona   ||   ',   '   ||
    p_titulacio, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifProfessor',   p_idPersona   ||   ',   '   ||
    p_titulacio, RSP);
  WHEN noExisteixProfessor THEN
  RSP:= 'ERROR: L'identificador introduït no correspon a cap professor';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifProfessor',   p_idPersona   ||   ',   '   ||
    p_titulacio, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut modificar';
```

```
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifProfessor',   p_idPersona   ||   ',   '   ||
    p_titulacio, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en modificar les dades del professor';
  INSERT INTO Log (data, procediment, entrada, sortida)
  VALUES (SYSDATE, 'modifProfessor', p_idPersona || ', ' || p_titulacio,
    RSP);
END modifProfessor;
```

(índex)

```
/*********************************************************************
Procediment que gestiona la modificació de les dades d'un curs a la
taula Curs a partir del seu identificador
Nom Procediment: modifCurs
Paràmetres d'entrada: p_idCurs, p_nomCurs
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE modifCurs (
  p_idCurs IN Number,
  p_nomCurs IN Varchar2,
  p_nombreGrups IN Number,
  RSP OUT Varchar2
)
IS
  noCurs Number;
  duplicat Number;
  campsNuls EXCEPTION;
  noExisteixCurs EXCEPTION;
  cursDuplicat EXCEPTION;
  cursNoValid EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF(p_nomCurs IS NULL)
    OR (p_nombreGrups IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que existeixi el curs a què fa referència
  l'identificador*/
  SELECT COUNT(*) INTO noCurs
  FROM Curs
  WHERE idCurs = p_idCurs;
  IF (noCurs = 0)
  THEN
    RAISE noExisteixCurs;
  END IF;
/*Es verifica que el nom del curs que es modifiqui no estigui duplicat*/
  SELECT COUNT(*) INTO duplicat
  FROM Curs
  WHERE nomCurs = p_nomCurs AND nombreGrups = p_nombreGrups;
  IF (duplicat > 0)
  THEN
    RAISE cursDuplicat;
  END IF;
/*Es verifica que s'introdueix un valor correcte en el nom del curs que
  es modifiqui*/
  IF (p_nomCurs NOT LIKE '1r. ESO' AND p_nomCurs NOT LIKE  '2n. ESO' AND
    p_nomCurs NOT LIKE '3r. ESO' AND p_nomCurs NOT LIKE '4t. ESO' AND
    p_nomCurs NOT LIKE '1r. Batxillerat' AND p_nomCurs NOT LIKE '2n.
    Batxillerat')
  THEN
    RAISE cursNoValid;
  END IF;
/*Es modifiquen les dades del curs a la taula Curs*/
  UPDATE Curs
  SET  nomCurs = p_nomCurs,
       nombreGrups = p_nombreGrups
```

```
  WHERE idCurs = p_idCurs;
/*S'actualitza la taula de sancions per curs i any si ha hagut
 modificació en el nom del curs*/
  UPDATE SancionsCursAny
  SET  nomCurs = p_nomCurs
  WHERE idCurs = p_idCurs;
/*S'actualitza la taula de mitjana de sancions per curs si ha hagut
 modificació en el nom del curs*/
  UPDATE AvgSancionsCurs
  SET  nomCurs = p_nomCurs
  WHERE idCurs = p_idCurs;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifCurs', p_idCurs || ', ' ||p_nomCurs || ', '
    ||p_nombreGrups, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifCurs', p_idCurs || ', ' ||p_nomCurs || ', '
    ||p_nombreGrups, RSP);
  WHEN noExisteixCurs THEN
  RSP:= 'ERROR: L'identificador introduït no correspon a cap curs';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifCurs', p_idCurs || ', ' ||p_nomCurs || ', '
    ||p_nombreGrups, RSP);
  WHEN cursDuplicat THEN
  RSP:= 'ERROR: Aquest curs ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifCurs', p_idCurs || ', ' ||p_nomCurs || ', '
    ||p_nombreGrups, RSP);
  WHEN cursNoValid THEN
  RSP:= 'ERROR: El nom del curs no és correcte';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifCurs', p_idCurs || ', ' ||p_nomCurs || ', '
    ||p_nombreGrups, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut modificar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifCurs', p_idCurs || ', ' ||p_nomCurs || ', '
    ||p_nombreGrups, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en modificar les dades del curs';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifCurs', p_idCurs || ', ' ||p_nomCurs || ', '
    ||p_nombreGrups, RSP);
END modifCurs;
```

(índex)

```
/*************************************************************************
Procediment que gestiona la modificació de les dades d'una assignatura a
la taula Assignatura a partir dels seus identificadors.
Nom Procediment: modifAssignatura
Paràmetres d'entrada: p_idCurs, p_idAssignatura, p_nomAssignatura
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE modifAssignatura (
  p_idCurs IN Number,
  p_idAssignatura IN Number,
  p_nomAssignatura IN Varchar2,
  RSP OUT Varchar2
)
IS
  noAssignatura Number;
  duplicada Number;
  campsNuls EXCEPTION;
  noExisteixAssignatura EXCEPTION;
  assigDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls */
  IF (p_idCurs IS NULL)
    OR (p_idAssignatura IS NULL)
    OR (p_nomAssignatura IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que l'assignatura que es vol modificar existeixi*/
  SELECT COUNT(*) INTO noAssignatura
  FROM Assignatura
  WHERE idCurs = p_idCurs AND idAssignatura = p_idAssignatura;
  IF (noAssignatura= 0)
  THEN
    RAISE noExisteixAssignatura;
  END IF;
/*Es verifica que l'assignatura que es modifiqui no estigui duplicada*/
  SELECT COUNT(*) INTO duplicada
  FROM Assignatura
  WHERE idCurs = p_idCurs AND idAssignatura = p_idAssignatura AND
    nomAssignatura = p_nomAssignatura;
  IF (duplicada> 0)
  THEN
    RAISE assigDuplicada;
  END IF;
/*Es modifiquen les dades de l'assignatura a la taula Assignatura */
  UPDATE Assignatura
  SET  nomAssignatura = p_nomAssignatura
  WHERE idCurs = p_idCurs AND idAssignatura = p_idAssignatura;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
```

```
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN noExisteixAssignatura THEN
  RSP:= 'ERROR: L''assignatura indicada no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN assigDuplicada THEN
  RSP:= 'ERROR: Aquesta assignatura ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut modificar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en modificar les dades de l''assignatura';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAssignatura',   p_idCurs   ||   ',   '   ||
    p_idAssignatura || ', ' || p_nomAssignatura, RSP);
END modifAssignatura;
```

(índex)

```
/*********************************************************************
Procediment que gestionala modificació de les dades d'una amonestació a
la taula Amonestacio a partir del seu identificador.
Nom Procediment: modifAmonestacio
Paràmetres d'entrada: p_idAmonestacio, p_tipus, p_descripcio, p_gravetat
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE modifAmonestacio (
  p_idAmonestacio IN Number,
  p_tipus IN Varchar2,
  p_descripcio IN Varchar2,
  p_gravetat IN Varchar2,
  RSP OUT Varchar2
)
IS
  noAmonestacio NUMBER;
  campsNuls EXCEPTION;
  noExisteixAmonestacio EXCEPTION;
  gravetatIncorrecta EXCEPTION;
  amonestacioDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls*/
  IF(p_tipus IS NULL)
    OR (p_descripcio IS NULL)
    OR (p_gravetat IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que l'amonestació que es vol modificar existeixi*/
  SELECT COUNT(*) INTO noAmonestacio
  FROM Amonestacio
  WHERE idAmonestacio = p_idAmonestacio;
  IF (noAmonestacio = 0)
  THEN
    RAISE noExisteixAmonestacio;
  END IF;
/*Es verifica que, si es modifica, el tipus d'amonestació no es dupliqui
 */
  SELECT COUNT(*) INTO noAmonestacio
  FROM Amonestacio
  WHERE tipus = p_tipus AND descripcio = p_descripcio AND gravetat =
    p_gravetat;
  IF (noAmonestacio> 0)
  THEN
    RAISE amonestacioDuplicada;
  END IF;
/*Es verifica que, si es modifica, s'introdueixen valors correctes en el
 camp gravetat*/
  IF (p_gravetat NOT LIKE 'Lleu' AND p_gravetat NOT LIKE 'Greu' AND
     p_gravetat NOT LIKE 'Molt greu')
  THEN
    RAISE gravetatIncorrecta;
  END IF;
/*Es modifiquen les dades de l'amonestacio a la taula Amonestacio */
  UPDATE Amonestacio
  SET  tipus = p_tipus,
       descripcio = p_descripcio,
```

```
      gravetat = p_gravetat
  WHERE idAmonestacio = p_idAmonestacio;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAmonestacio',   p_idAmonestacio   ||   ',   '
    ||p_tipus || ', ' || p_descripcio || ', ' || p_gravetat, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAmonestacio',   p_idAmonestacio   ||   ',   '
    ||p_tipus || ', ' || p_descripcio || ', ' || p_gravetat, RSP);
  WHEN noExisteixAmonestacio THEN
  RSP:=   'ERROR:   L''identificador   indicat   no   correspon   a   cap
         amonestació';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAmonestacio',   p_idAmonestacio   ||   ',   '
    ||p_tipus || ', ' || p_descripcio || ', ' || p_gravetat, RSP);
  WHEN amonestacioDuplicada THEN
  RSP:= 'ERROR: El tipus d''amonestació ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAmonestacio',   p_idAmonestacio   ||   ',   '
    ||p_tipus || ', ' || p_descripcio || ', ' || p_gravetat, RSP);
  WHEN gravetatIncorrecta THEN
  RSP:= 'ERROR: La gravetat de l''amonestació només pot ser Lleu, Greu o
        Molt greu';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAmonestacio',   p_idAmonestacio   ||   ',   '
    ||p_tipus || ', ' || p_descripcio || ', ' || p_gravetat, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut modificar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAmonestacio',   p_idAmonestacio   ||   ',   '
    ||p_tipus || ', ' || p_descripcio || ', ' || p_gravetat, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en modificar les dades de l''amonestació';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES   (SYSDATE,   'modifAmonestacio',   p_idAmonestacio   ||   ',   '
    ||p_tipus || ', ' || p_descripcio || ', ' || p_gravetat, RSP);
END modifAmonestacio;
```

(índex)

```
/*********************************************************************
Procediment que gestionala modificació de les dades d'una sanció a la
taula Sancio a partir del seu identificador.
Nom Procediment: modifSancio
Paràmetres d'entrada: p_idSancio, p_tipus, p_activacio
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*********************************************************************/
CREATE OR REPLACE PROCEDURE modifSancio (
  p_idSancio IN NUMBER,
  p_tipus IN Varchar2,
  p_activacio IN Varchar2,
  RSP OUT Varchar2
)
IS
  noSancio NUMBER;
  campsNuls EXCEPTION;
  noExisteixSancio EXCEPTION;
  sancioDuplicada EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls*/
  IF (p_tipus IS NULL)
    OR (p_activacio IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que la sanció que es vol modificar existeixi*/
  SELECT COUNT(*) INTO noSancio
  FROM Sancio
  WHERE idSancio = p_idSancio;
  IF (noSancio = 0)
  THEN
    RAISE noExisteixSancio;
  END IF;
/*Es verifica que el tipus de sanció no es dupliqui*/
  SELECT COUNT(*) INTO noSancio
  FROM Sancio
  WHERE tipus = p_tipus AND activacio = p_activacio;
  IF (noSancio > 0)
  THEN
    RAISE sancioDuplicada;
  END IF;
/*Es modifiquen les dades de la sanció a la taula Sancio*/
  UPDATE Sancio
  SET  tipus = p_tipus,
       activacio = p_activacio
  WHERE idSancio = p_idSancio;
  RSP:= 'OK';
/*S'actualitza la taula log */
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifSancio', p_idSancio || ', ' || p_tipus || ', '
    || p_activacio, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
```

```
    VALUES (SYSDATE, 'modifSancio', p_idSancio || ', ' ||p_tipus || ', '
    || p_activacio, RSP);
  WHEN noExisteixSancio THEN
  RSP:= 'ERROR: L''identificador indicat no correspon a cap sanció';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifSancio', p_idSancio || ', ' ||p_tipus || ', '
    || p_activacio, RSP);
  WHEN sancioDuplicada THEN
  RSP:= 'ERROR: El tipus de sanció ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifSancio', p_idSancio || ', ' ||p_tipus || ', '
    || p_activacio, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut modificar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifSancio', p_idSancio || ', ' ||p_tipus || ', '
    || p_activacio, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en modificar les dades de la sanció';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'modifSancio', p_idSancio || ', ' || p_tipus || ', '
    || p_activacio, RSP);
END modifSancio;
```

(índex)

```
/*************************************************************************
Procediment que gestiona la modificació de les dades d'una entrada del
calendari escolar a la taula CalendariEscolar a partir del seu
identificador.
Nom Procediment: modifCalendariEscolar
Paràmetres d'entrada: p_idCalendari, p_nomCalendari, p_data, p_curs,
p_assignatura, p_dia, p_hora
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE modifCalendariEscolar (
  p_idCalendari IN Number,
  p_nomCalendari IN Varchar2,
  p_data IN Date,
  p_curs IN Number,
  p_assignatura IN Number,
  p_dia IN Number,
  p_hora IN Number,
  RSP OUT Varchar2
)
IS
  noCalendari Number;
  festiu Number;
  noAssignatura Number;
  noHora Number;
  noDuplicat Number;
  noData Number;
  campsNuls EXCEPTION;
  noExisteixCalendari EXCEPTION;
  diaFestiu EXCEPTION;
  noExisteixAssignatura EXCEPTION;
  noExisteixHoraLectiva EXCEPTION;
  calendariDuplicat EXCEPTION;
BEGIN
/*Es verifica que es compleixen les restriccions dels camps no nuls*/
  IF (p_nomCalendari IS NULL)
    OR (p_data IS NULL)
    OR (p_curs IS NULL)
    OR (p_assignatura IS NULL)
    OR (p_dia IS NULL)
    OR (p_hora IS NULL)
  THEN
    RAISE campsNuls;
  END IF;
/*Es verifica que el registre de CalendariEscolar que es vol modificar
 existeixi*/
  SELECT COUNT(*) INTO noCalendari
  FROM CalendariEscolar
  WHERE idCalendari = p_idCalendari;
  IF (noCalendari = 0)
  THEN
    RAISE noExisteixCalendari;
  END IF;
/*Es verifica que si es modifica el dia, aquest no sigui festiu*/
  SELECT COUNT(*)INTO festiu
  FROM Dates
  WHERE festiu = 'S' and data = p_data;
  IF (festiu > 0)
```

```
  THEN
    RAISE diaFestiu;
  END IF;
/*Es verifica que si es modifica l'assignatura, aquesta existeixi*/
  SELECT COUNT(*) INTO noAssignatura
  FROM Assignatura
  WHERE idCurs = p_curs AND idAssignatura = p_assignatura;
  IF (noAssignatura = 0)
  THEN
    RAISE noExisteixAssignatura;
  END IF;
/*Es verifica que si es modifica l'hora lectiva, aquesta existeixi*/
  SELECT COUNT(*) INTO noHora
  FROM HoraLectiva
  WHERE idDia = p_dia AND horaDia = p_hora;
  IF (noHora= 0)
  THEN
    RAISE noExisteixHoraLectiva;
  END IF;
/*Es verifica que el registre resultant de la modificació no estigui
 duplicat */
  SELECT COUNT(*) INTO noDuplicat
  FROM CalendariEscolar
  WHERE data = p_data AND curs = p_curs AND assignatura = p_assignatura
        AND dia = p_dia AND hora = p_hora;
  IF (noDuplicat> 0)
  THEN
    RAISE calendariDuplicat;
  END IF;
/*Si la data indicada no hi és a la taula Dates, s'hi insereix*/
  SELECT COUNT(*)INTO noData
  FROM Dates
  WHERE data = p_data;
  IF (noData= 0)
  THEN
    INSERT INTO Dates (data, festiu) VALUES (p_data, 'N');
  END IF;
/*Es modifiquen les dades del registre del calendari a la taula
 CalendariEscolar*/
  UPDATE CalendariEscolar
  SET  nomCalendari = p_nomCalendari,
       data = p_data,
       curs = p_curs,
       assignatura = p_assignatura,
       dia = p_dia,
       hora = p_hora
  WHERE idCalendari = p_idCalendari;
  RSP:= 'OK';
/*S'actualitza la taula log*/
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari || ', ' || p_data || ', ' || p_curs || ', ' ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  COMMIT;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN campsNuls THEN
  RSP:= 'ERROR: Hi ha camps que no poden ser nuls';
  INSERT INTO Log (data, procediment, entrada, sortida)
```

```
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  WHEN noExisteixCalendari THEN
  RSP:= 'ERROR: No existeix cap registre a Calendari Escolar amb aquest
        identificador';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  WHEN diaFestiu  THEN
  RSP:= 'ERROR: La data indicada correspon a un dia festiu';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  WHEN noExisteixAssignatura THEN
  RSP:=  'ERROR:  L''identificador  del  curs  i/o  de  l''assignatura
    introduït no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  WHEN noExisteixHoraLectiva THEN
  RSP:= 'ERROR: L''hora i/o dia indicat no existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  WHEN calendariDuplicat THEN
  RSP:= 'ERROR: El registre de Calendari Escolar indicat ja existeix';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  WHEN storage_error THEN
  RSP:= 'ERROR: El registre no s''ha pogut modificar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
  WHEN others THEN
  RSP:=  'ERROR:  Error  genèric  en  modificar  les  dades  del  calendari
    escolar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES  (SYSDATE,  'modifCalendariEscolar',  p_idCalendari  ||  ',  '
    ||p_nomCalendari  ||  ',  '  ||  p_data  ||  ',  '  ||  p_curs  ||  ',  '  ||
    p_assignatura || ', ' || p_dia || ', ' || p_hora, RSP);
END modifCalendariEscolar;
```

(índex)

```
/*************************************************************************
Procediment de consulta del llistat de totes les amonestacions
imposades, sense cap filtratge, indicant-ne la seva informació bàsica i
ordenat per la data i l'hora de l'amonestació
Nom Procediment: llistatAmonestacionsImposades
Paràmetres d'entrada: cap
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE llistatAmonestacionsImposades(
  RSP OUT Varchar2
)
IS
  nombreFiles NUMBER;
/*Cursor del llistat*/
  CURSOR c_amonestacions IS
  SELECT     (al.nom || ' ' ||al.cognom1 || ' ' || al.cognom2) alumne,
       a.tipus,
       a.gravetat,
       em.data,
       em.horaAmonestacio,
       c.nomCurs,
       ass.nomAssignatura,
       (pr.nom ||' '||pr.cognom1||' '||pr.cognom2) professor,
       em.comunicacioPares
  FROM   Persona al, Persona pr, AmonestacionsEmeses em, Amonestacio a,
         Assignatura ass, Curs c
  WHERE  al.idPersona = em.alumne AND em.amonestacio = a.idAmonestacio
         AND em.assignatura = ass.idAssignatura AND em.curs = c.idCurs
         AND ass.idCurs = c.idCurs AND em.professor = pr.idPersona
  ORDER BY em.data, em.horaAmonestacio;
/*Tipus llistat*/
  cur_amonestacions      c_amonestacions%ROWTYPE;
BEGIN
/*Es crea la capçalera del llistat*/
  DBMS_OUTPUT.PUT_LINE     (rpad('Alumne',28,'     ')||'     '||rpad('Tipus
    d''amonestació',26,'      ')||'      '||rpad('Gravetat',11,'      ')||'
    '||rpad('Data',10,'      ')      ||'      '||rpad('Hora',6,'      ')||'
    '||rpad('Curs',15,'    ')||'    '||rpad('Assignatura',20,'    ')||'    '
    ||rpad('Professor',28,' ')||' ' ||rpad('Comunicat pares',16,' '));
  DBMS_OUTPUT.PUT_LINE     (rpad('-',28,'-')||'     '||rpad('-',26,'-')||'
    '||rpad('-',11,'-')||' '||rpad('-',10,'-') ||' '||rpad('-',6,'-')||'
    '||rpad('-',15,'-')||' '||rpad('-',20,'-')||' ' ||rpad('-',28,'-')||'
    ' ||rpad('-',16,'-'));
/*S'executa el cursor amb les dades del llistat*/
  FOR cur_amonestacions IN c_amonestacions
    LOOP
      DBMS_OUTPUT.PUT_LINE     (rpad(cur_amonestacions.alumne,28,  '  ')||
    '      '||rpad(cur_amonestacions.tipus,26,'      ')||      '      '||
    rpad(cur_amonestacions.gravetat,11,'                              ')||'
    '||rpad(cur_amonestacions.data,10,          '          ')          ||'
    '||rpad(cur_amonestacions.horaAmonestacio,6,'          ')||          '
    '||rpad(cur_amonestacions.nomCurs,15,              '             ')||'
    '||rpad(cur_amonestacions.nomAssignatura,20,'      ')||      '      '
    ||rpad(cur_amonestacions.professor,28,    '    ')||    '    '    ||rpad('
    '||cur_amonestacions.comunicacioPares,16, ' '));
      nombreFiles:=c_amonestacions%rowcount;
    END LOOP;
```

```
  IF (nombreFiles>0) THEN
    DBMS_OUTPUT.NEW_LINE;
    DBMS_OUTPUT.PUT_LINE('Total     d''amonestacions     imposades     =
    '||to_char(nombreFiles));
  END IF;
  RSP:= 'OK';
/*S'actualitza la taula log*/
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'llistatAmonestacionsImposades', ' ', RSP);
  RETURN;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN no_data_found THEN
  RSP:= 'ERROR: No hi ha dades per ensenyar';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'llistatAmonestacionsImposades', ' ', RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en llistar les dades sol•licitades';
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES (SYSDATE, 'llistatAmonestacionsImposades', ' ', RSP);
END llistatAmonestacionsImposades;
```

(índex)

```
/*************************************************************************
Procediment de consulta del llistat de tots els alumnes d'un curs,
indicant-ne la seva informació bàsica
Nom Procediment: llistatAlumnesCurs
Paràmetres d'entrada: nomCurs
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE llistatAlumnesCurs (
  p_nomCurs INVarchar2,
  RSP OUT Varchar2
)
IS
  nombreFiles NUMBER;
  cursNoValid EXCEPTION;
/*Cursor del llistat*/
  CURSOR c_alumnesCurs IS
  SELECT     (p.nom || ' ' ||p.cognom1 || ' ' || p.cognom2) alumne,
       p.docIdentificatiu,
       a.numExpedient,
       a.dataNaixement,
       p.adreça,
       m.codiPostal,
       m.nomMunicipi,
       cm.nomComarca,
       pr.nomProvincia,
       p.telefon,
       p.eMail,
       a.nomPare,
       a.nomMare,
       a.anyIngres,
       c.nomCurs
  FROM Persona p, Alumne a, Municipi m, Comarca cm, Provincia pr,
    Matriculat mt, Curs c
  WHERE      p.idPersona = a.idPersona AND p.municipi = m.idMunicipi AND
    m.comarca = cm.idComarca AND cm.provincia = pr.idProvincia AND
    a.idPersona = mt.alumne AND mt.curs = c.idCurs AND c.nomCurs =
    p_nomCurs
  ORDER BY p.cognom1, p.cognom2, p.nom;
/*Tipus llistat*/
  cur_alumnesCurs      c_alumnesCurs%ROWTYPE;
BEGIN
/*Es verifica que s'introdueix un valor correcte en el nom del curs*/
  IF (p_nomCurs NOT LIKE '1r. ESO' AND p_nomCurs NOT LIKE '2n. ESO' AND
    p_nomCurs NOT LIKE '3r. ESO' AND p_nomCurs NOT LIKE '4t. ESO' AND
    p_nomCurs NOT LIKE '1r. Batxillerat' AND p_nomCurs NOT LIKE '2n.
    Batxillerat')
  THEN
    RAISE cursNoValid;
  END IF;
/*Es crea la capçalera del llistat*/
  DBMS_OUTPUT.PUT_LINE  (rpad('Curs',15,'  ')||'  '||rpad('Alumne',26,'
    ')||'             '||rpad('Doc.identificació',17,'             ')||'
    '||rpad('Núm.expedient',13,' ')||' '||rpad('Data de naixement',17,'
    ')||'  '||rpad('Adreça',25,'  ')||'  '||rpad('C.Postal',8,'  ')||'
    '||rpad('Municipi',20,'  ')||'  '  ||rpad('Comarca',20,'  ')||'  '
    ||rpad('Província',11,'  ') ||'  '  ||rpad('Telèfon',11,'  ') ||'  '
```

```
   ||rpad('eMail',20,' ') ||' ' ||rpad('Ingrés',6,' ') ||' ' ||rpad('Nom
   del pare',12,' ') ||' ' ||rpad('Nom de la mare',14,' '));
  DBMS_OUTPUT.PUT_LINE    (rpad('-',15,'-')||'    '||rpad('-',26,'-')||'
   '||rpad('-',17,'-')||' '||rpad('-',13,'-') ||' '||rpad('-',17,'-')||'
   '||rpad('-',25,'-')||' '||rpad('-',8,'-')||' ' ||rpad('-',20,'-')||'
   ' ||rpad('-',20,'-') ||' ' ||rpad('-',11,'-') ||' ' ||rpad('-',11,'-
   ') ||' ' ||rpad('-',20,'-') ||' ' ||rpad('-',6,'-') ||' ' ||rpad('-
   ',12,'-') ||' ' ||rpad('-',14,'-'));
/*S'executa el cursor amb les dades del llistat*/
  FOR cur_alumnesCurs IN c_alumnesCurs
   LOOP
      DBMS_OUTPUT.PUT_LINE   (rpad(cur_alumnesCurs.nomCurs,15,'   ')||'
   '||rpad(cur_alumnesCurs.alumne,26,'                          ')||'
   '||rpad(cur_alumnesCurs.docIdentificatiu,17,'                ')||'
   '||rpad(cur_alumnesCurs.numExpedient,13,'                    ')||'
   '||rpad(cur_alumnesCurs.dataNaixement,17,'                   ')||'
   '||rpad(cur_alumnesCurs.adreça,25,'                          ')||'
   '||rpad(cur_alumnesCurs.codiPostal,8,'                       ')||'
   '||rpad(cur_alumnesCurs.nomMunicipi,20,'                     ')||'
   '||rpad(cur_alumnesCurs.nomComarca,20,'                      ')||'
   '||rpad(cur_alumnesCurs.nomProvincia,11,'                    ')||'
   '||rpad(cur_alumnesCurs.telefon,11,'                         ')||'
   '||rpad(cur_alumnesCurs.eMail,20,'                           ')||'
   '||rpad(cur_alumnesCurs.anyIngres,6,'                        ')||'
   '||rpad(cur_alumnesCurs.nomPare,12,'                         ')||'
   '||rpad(cur_alumnesCurs.nomMare,14,' '));
      nombreFiles:=c_alumnesCurs%rowcount;
   END LOOP;
IF (nombreFiles > 0) THEN
      DBMS_OUTPUT.NEW_LINE;
      DBMS_OUTPUT.PUT_LINE('Total   d''alumnes   del   curs:   '   ||
   to_char(nombreFiles));
   END IF;
  RSP:= 'OK';
/*S'actualitza la taula log*/
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES (SYSDATE, 'llistatAlumnesCurs', p_nomCurs, RSP);
  RETURN;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN cursNoValid THEN
  RSP:= 'ERROR: El nom del curs no és correcte';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES (SYSDATE, 'llistatAlumnesCurs', p_nomCurs, RSP);
  WHEN no_data_found THEN
  RSP:= 'ERROR: No hi ha dades per ensenyar';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES (SYSDATE, 'llistatAlumnesCurs', p_nomCurs, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en llistar les dades sol•licitades';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES (SYSDATE, 'llistatAlumnesCurs', p_nomCurs, RSP);
END llistatAlumnesCurs;
```

(índex)

```
/*************************************************************************
Procediment de consulta del llistat de tots els tipus d'amonestacions o
de tots els tipus de sancions disponibles
Nom Procediment: llistatAmonestacionsSancions
Paràmetre d'entrada: p_amonestacio_sancio (permet triar entre un llistat
d'Amonestacions o un llistat de Sancions)
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACE PROCEDURE llistatAmonestacionsSancions (
  p_amonestacio_sancio IN Varchar2,
  RSP OUT Varchar2
)
IS
  nombreFiles NUMBER;
  seleccioNoValida EXCEPTION;
/*Cursor per al llistat d'amonestacions*/
  CURSOR c_amonestacions IS
  SELECT *
  FROM   Amonestacio
  ORDER BY idAmonestacio;
/*Cursor per al llistat de sancions*/
  CURSOR c_sancions IS
  SELECT *
  FROM   Sancio
  ORDER BY idSancio;
/*Tipus llistat*/
  cur_amonestacions      c_amonestacions%ROWTYPE;
  cur_sancions           c_sancions%ROWTYPE;
BEGIN
/*Es verifica que s'introdueix un valor correcte en la selecció del
  llistat*/
  IF(p_amonestacio_sancio      NOT      LIKE      'Amonestacions'      AND
    p_amonestacio_sancio NOT LIKE 'Sancions')
  THEN
    RAISE seleccioNoValida;
  END IF;
/*S'executa el cursor segons el tipus de llistat seleccionat*/
  CASE
    WHEN p_amonestacio_sancio = 'Amonestacions' THEN
/*Es crea la capçalera del llistat*/
    DBMS_OUTPUT.PUT_LINE    (rpad('Tipus    d''amonestació',55,'    ')||'
    '||rpad('Descripció        de        l''amonestació',70,'        ')||'
    '||rpad('Gravetat',15,' '));
    DBMS_OUTPUT.PUT_LINE    (rpad('-',55,'-')||'    '||rpad('-',70,'-')||'
    '||rpad('-',15,'-'));
/*S'executa el cursor amb les dades del llistat*/
    FOR cur_amonestacions IN c_amonestacions
       LOOP
         DBMS_OUTPUT.PUT_LINE(rpad  (cur_amonestacions.tipus,55,'   ')||'
    '||rpad           (cur_amonestacions.descripcio,70,'            ')||'
    '||rpad(cur_amonestacions.gravetat,15, ' '));
         nombreFiles:=c_amonestacions%rowcount;
       END LOOP;
    IF (nombreFiles>0) THEN
       DBMS_OUTPUT.NEW_LINE;
       DBMS_OUTPUT.PUT_LINE('Total    tipus    d''amonestacions:    '    ||
    to_char(nombreFiles));
```

```
   END IF;
   WHEN p_amonestacio_sancio = 'Sancions' THEN
/*Es crea la capçalera del llistat*/
   DBMS_OUTPUT.PUT_LINE     (rpad('Tipus     de     sanció',55,'     ')||'
   '||rpad('Motiu d''imposició',70,' '));
   DBMS_OUTPUT.PUT_LINE (rpad('-',55,'-')||' '||rpad('-',70,'-'));
/*S'executa el cursor amb les dades del llistat*/
   FOR cur_sancions IN c_sancions
     LOOP
       DBMS_OUTPUT.PUT_LINE      (rpad(cur_sancions.tipus,55,'      ')||'
   '||rpad(cur_sancions.activacio,70,' '));
       nombreFiles:=c_sancions%rowcount;
     END LOOP;
   IF (nombreFiles>0) THEN
       DBMS_OUTPUT.NEW_LINE;
       DBMS_OUTPUT.PUT_LINE('Total    tipus    de    sancions:    '    ||
   to_char(nombreFiles));
   END IF;
  END CASE;
  RSP:= 'OK';
/*S'actualitza la taula log*/
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES(SYSDATE, 'llistatAmonestacionsSancions', p_amonestacio_sancio,
   RSP);
  RETURN;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN seleccioNoValida THEN
  RSP:= 'ERROR: La selecció del llistat no és correcta';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES           (SYSDATE,           'llistatAmonestacionsSancions',
   p_amonestacio_sancio, RSP);
  WHEN no_data_found THEN
  RSP:= 'ERROR: No hi ha dades per ensenyar';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES           (SYSDATE,           'llistatAmonestacionsSancions',
   p_amonestacio_sancio, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en llistar les dades sol•licitades';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES           (SYSDATE,           'llistatAmonestacionsSancions',
   p_amonestacio_sancio, RSP);
END llistatAmonestacionsSancions;
```

(índex)

```
/*************************************************************************
Procediment de consulta del llistat de totes les amonestacions o de
totes les sancions d'un alumne, ordenat per data.
Nom Procediment: llistatAmonSancionsAlumne
Paràmetre d'entrada: p_amonestacio_sancio (permet triar entre un llistat
d'Amonestacions o un llistat de Sancions), p_alumne
Sortida: RSP
Retorna "OK" si l'execució finalitza amb èxit; retorna "ERROR: Tipus
d'error" si fracassa.
*************************************************************************/
CREATE OR REPLACEPROCEDURE llistatAmonSancionsAlumne (
  p_amonestacio_sancio IN Varchar2,
  p_alumne IN Number,
  RSP OUT Varchar2
)
IS
  nombreFiles NUMBER;
  noAlumne Number;
  seleccioNoValida EXCEPTION;
  noExisteixAlumne EXCEPTION;
/*Cursor per al llistat d'amonestacions*/
  CURSOR c_amonestacions IS
  SELECT(p.nom || ' ' || p.cognom1 || ' ' || p.cognom2) alumne,
       tipus,
       descripcio,
       gravetat,
       data
  FROM Persona p, Alumne al, AmonestacionsEmeses em, Amonestacio am
  WHERE  p.idPersona  =  al.idPersona  AND  al.idPersona  =  em.alumne  AND
         em.amonestacio = am.idAmonestacio AND al.idPersona = p_alumne
  ORDER BY data;
/*Cursor per al llistat de sancions*/
  CURSOR c_sancions IS
  SELECT     (p.nom || ' ' || p.cognom1 || ' ' || p.cognom2) alumne,
       tipus,
       activacio,
       data
  FROM  Persona p, Alumne al, SancionsEmeses em, Sancio s
  WHERE  p.idPersona  =  al.idPersona  AND  al.idPersona  =  em.alumne  AND
         em.sancio = s.idSancio AND al.idPersona = p_alumne
  ORDER BY data;
/*Tipus llistat*/
  cur_amonestacions    c_amonestacions%ROWTYPE;
  cur_sancions         c_sancions%ROWTYPE;
BEGIN
/*Es verifica que s'introdueix un valor correcte en la selecció del
  llistat*/
  IF(p_amonestacio_sancio      NOT      LIKE      'Amonestacions'      AND
p_amonestacio_sancio NOT LIKE 'Sancions')
  THEN
    RAISE seleccioNoValida;
  END IF;
/*Es verifica que existeixi l'alumne a què es fa referència*/
  SELECTCOUNT(*) INTO noAlumne
  FROM Alumne
  WHERE idPersona = p_alumne;
  IF (noAlumne = 0)
  THEN
    RAISE noExisteixAlumne;
```

```
  END IF;
/*S'executa el cursor segons el tipus de llistat seleccionat*/
  CASE
    WHEN p_amonestacio_sancio = 'Amonestacions' THEN
/*Es crea la capçalera del llistat*/
    DBMS_OUTPUT.PUT_LINE    (rpad('Alumne',35,'    ')||'    '||rpad('Tipus
d''amonestació',26,' ') ||' '||rpad('Motiu de l''amonestació',45,' ')||'
'||rpad('Gravetat',15,' ') ||' '||rpad('Data',10,' '));
    DBMS_OUTPUT.PUT_LINE  (rpad('-',35,'-')||'  '||rpad('-',26,'-')  ||'
'||rpad('-',45,'-')||' '||rpad('-',15,'-') ||' '||rpad('-',10,'-'));
/*S'executa el cursor amb les dades del llistat*/
    FOR cur_amonestacions IN c_amonestacions
      LOOP
        DBMS_OUTPUT.PUT_LINE  (rpad(cur_amonestacions.alumne,35,  '  ')||'
'||rpad(cur_amonestacions.tipus,26,                '              ')||'
'||rpad(cur_amonestacions.descripcio,45,             '            ')||'
'||rpad(cur_amonestacions.gravetat,15,              '             ')||'
'||rpad(cur_amonestacions.data,10,' '));
       nombreFiles:=c_amonestacions%rowcount;
      END LOOP;
    IF (nombreFiles>0) THEN
      DBMS_OUTPUT.NEW_LINE;
      DBMS_OUTPUT.PUT_LINE('Total    d''amonestacions    de    l''alumne
'||cur_amonestacions.alumne||': ' || to_char(nombreFiles));
    END IF;
    WHEN p_amonestacio_sancio = 'Sancions' THEN
/*Es crea la capçalera del llistat*/
      DBMS_OUTPUT.PUT_LINE  (rpad('Alumne',35,'  ')||'  '||rpad('Tipus  de
sanció',55,'   ')   ||'   '||rpad('Motiu   de   la   sanció',70,'   ')||'
'||rpad('Data',10,' '));
      DBMS_OUTPUT.PUT_LINE  (rpad('-',35,'-')||'  '||rpad('-',55,'-')  ||'
'||rpad('-',70,'-')||' '||rpad('-',10,'-'));
/*S'executa el cursor amb les dades del llistat*/
    FOR cur_sancions IN c_sancions
      LOOP
        DBMS_OUTPUT.PUT_LINE    (rpad(cur_sancions.alumne,35,'    ')    ||'
'||rpad(cur_sancions.tipus,55,              '             ')            ||'
'||rpad(cur_sancions.activacio,70,           '          ')           ||'
'||rpad(cur_sancions.data,10, ' '));
             nombreFiles:=c_sancions%rowcount;
      END LOOP;
    IF (nombreFiles>0) THEN
      DBMS_OUTPUT.NEW_LINE;
      DBMS_OUTPUT.PUT_LINE('Total   de   sancions   de   l''alumne:   '   ||
to_char(nombreFiles));
    END IF;
  END CASE;
  RSP:= 'OK';
/*S'actualitza la taula log*/
  INSERT INTO Log (data, procediment, entrada, sortida)
    VALUES(SYSDATE,  'llistatAmonSancionsAlumne',  p_amonestacio_sancio||
', ' || p_alumne, RSP);
  RETURN;
/*Gestió d'excepcions*/
  EXCEPTION
  WHEN seleccioNoValida THEN
  RSP:= 'ERROR: La selecció del llistat no és correcta';
  INSERT INTO Log (data, procediment, entrada, sortida)
```

```
   VALUES (SYSDATE, 'llistatAmonSancionsAlumne', p_amonestacio_sancio ||
', ' || p_alumne, RSP);
  WHEN noExisteixAlumne THEN
  RSP:= 'ERROR: L''identificador introduït no correspon a cap alumne';
  INSERTINTO Log (data, procediment, entrada, sortida)
   VALUES (SYSDATE, 'llistatAmonSancionsAlumne', p_amonestacio_sancio ||
', ' || p_alumne, RSP);
  WHEN no_data_found THEN
  RSP:= 'ERROR: No hi ha dades per ensenyar';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES (SYSDATE, 'llistatAmonSancionsAlumne', p_amonestacio_sancio ||
', ' || p_alumne, RSP);
  WHEN others THEN
  RSP:= 'ERROR: Error genèric en llistar les dades sol•licitades';
  INSERT INTO Log (data, procediment, entrada, sortida)
   VALUES (SYSDATE, 'llistatAmonSancionsAlumne', p_amonestacio_sancio ||
', ' || p_alumne, RSP);
END llistatAmonSancionsAlumne;
```

(índex)