



# **GESTIÓ D'AMONESTACIONS I SANCIONS EN UN CENTRE EDUCATIU**

**Memòria de treball**

**Lluís Aristondo Arvidsson  
ETIG**

**Consultor: Jordi Ferrer Duran**

**Lliurament final**



## Índex de continguts

1 Capítol 1, Introducció.....	3
1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.....	3
1.2 Objectius del TFC.....	4
1.3 Productes obtinguts.....	4
1.4 Programari utilitzat per a l'elaboració del projecte.....	5
1.5 Breu descripció dels altres capítols de la memòria.....	5
2 Planificació del projecte.....	6
2.1 Planificació inicial del projecte.....	6
2.2 Modificacions de la planificació.....	7
3 Anàlisi dels requisits.....	9
3.1 Muntatge de la BD.....	10
4 Disseny de la BD.....	11
4.1 Taules.....	11
4.2 Vistes.....	16
4.3 Taules del mòdul estadístic.....	16
4.4 Vistes del mòdul estadístic.....	19
5 Scripts de la BD.....	20
5.1 Taules.....	20
5.2 Consultes.....	20
5.3 Funcions.....	20
5.4 Procediments.....	22
5.5 Procediments mòdul estadístic.....	27
5.6 Disparadors.....	28
6 Jocs de proves.....	35
7 Presentació virtual.....	35
8 Glossari.....	37
9 Bibliografia.....	38
10 Annexos.....	39
10.1 Presentació virtual.....	39
10.2 Scripts de la base de dades.....	39
10.3 Disseny de la BD.....	40

## 1 Capítol 1, Introducció.

### 1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.

L'objectiu d'aquest treball és dissenyar una base de dades que permeti gestionar de forma eficient la disciplina d'un centre educatiu. La base de dades s'ha de poder ampliar a futures funcionalitats, aprofitant tota la informació que hi tindrà referent a alumnes, cursos, professors i demés.

La base de dades ha de permetre introduir totes les dades d'alumnes i professors, com també importar dades d'altres sistemes d'informació com ara les bases de dades que contenen informació acadèmica dels alumnes i professors. Per aquest motiu la base de dades s'haurà d'adaptar a la informació existent en el sistema informàtic de la Conselleria d'Educació. És important aquest punt, ja que segons el nombre d'alumnes i professors d'un centre el volum de dades a introduir pot ser molt elevat.

La base de dades mantindrà informació d'un curs al següent sempre separant les informacions referents a cada un dels cursos acadèmics, ja que totes les sancions i amonestacions queden anul·lades en començar un nou curs. Només s'han de conservar dades d'anys anteriors a efectes d'estudis estadístics. A tal efecte és important poder estudiar els efectes que tenen sobre la disciplina les actuacions que es puguin dur a terme als centres, com ara veure si el fet d'instaurar un sistema de mediació escolar o xerrades amb alumnes sobre algun tema relacionat amb les sancions més habituals té repercussió sobre el nombre d'amonestacions i sancions posades.

El fet d'incloure els horaris dels diferents professors al sistema pot facilitar molt el seguiment de les amonestacions posades, i a detectar errades com ara una amonestació posada a un dia hora en la qual el professor no tenia classe amb el grup indicat.

Cada centre educatiu hauria de poder decidir qui serà encarregat d'introduir les informacions al sistema. Hi hauria d'haver un llistat d'usuaris amb contrasenya per poder accedir, amb permisos delimitats per a poder introduir només la informació que correspongui a cada un. Una possibilitat seria que cada professor pogués introduir les seves, una altra que els caps d'estudis i administratius fossin els encarregats de fer-ho. Tant en un cas com a l'altre hi hauria de quedar constància a la base de dades de quin usuari ha introduït cada informació i quan. En cas de modificació segons quins registres hi ha d'haver la possibilitat de recuperar la informació anterior a la modificació.

La normativa vigent contempla també que els desperfectes de les instal·lacions del centre ocasionats per accions deliberades o negligents dels alumnes han de ser reparats a compte del causant dels danys. Per la qual cosa en cas d'haver de reclamar una compensació econòmica acompanyant una sanció s'ha de poder indicar la quantitat, i s'ha de poder dur un seguiment dels pagaments efectuats.

Per a les sancions per acumulació d'amonestacions, s'han de tenir en compte uns aspectes: d'una banda les amonestacions que ja s'han sancionat no es poden tornar a comptabilitzar per a noves sancions. I d'altra banda, passat un determinat període de temps les amonestacions hauran prescrit i tampoc s'han de tenir en compte.

## 1.2 Objectius del TFC.

- Obtenir un producte que pugui ser utilitzat en un context real, per a facilitar la gestió de sancions i amonestacions en un centre educatiu.
- Aprofundir en el funcionament de les bases de dades relacionals.
- Conèixer i utilitzar el llenguatge de programació PL/SQL.
- Aconseguir realitzar de forma global un projecte, on a més de la part de continguts s'aconsegueixi cuidar la part de presentació de resultats i de la feina feta de cara a possibles clients de projectes.
- Elaborar de forma efectiva tota la documentació referent al producte de cara tant a la utilització per part de futurs usuaris com de cara a manteniment de la base de dades i futures ampliacions.

## 1.3 Productes obtinguts

### Pla de treball

Lliurat inicialment a la PAC 1, marcava les línies a seguir per aconseguir arribar al producte final. A mesura que ha avançat la feina s'ha vist que la planificació inicial no era la més adequada.

### Memòria

És el present document, on hi ha la major part de les explicacions i informacions relatives al progrés de la feina feta i explicacions referents al producte final: taules, vistes, procediments, etc.

### Document gràfic de disseny

És el gràfic que mostra un esquema amb totes les taules de la BD i les relacions entre elles. S'inclou als annexes dins aquest mateix document.

### Scripts de creació de la BD i d'inserció de registres

Els scripts per a la creació de taules i vistes, i per a la creació de disparadors, funcions i procediments seran inclosos al lliurament final en arxius d'extensió .sql. Aquests arxius venen llistats a l'apartat d'annexos de la memòria.

### Document amb les especificacions

A la planificació inicial es preveia un document explicatiu de tots els elements de la base de dades. Finalment els continguts d'aquest document s'han inclòs com a diferents apartats de la present memòria.

### Presentació virtual

La presentació virtual s'ha elaborat amb el OpenOffice Impress, i pretén facilitar una primera aproximació a la feina feta per elaborar la base de dades. Il·lustra diferents decisions que s'han pres en quant a disseny i funcionament de l'aplicació.

## Debat Virtual

Més que un producte elaborat, es pot considerar el debat virtual com una de les tasques pendents del projecte. Es tracta

### 1.4 Programari utilitzat per a l'elaboració del projecte

Per aquest projecte s'utilitzen bàsicament programes gratuïts de llicència lliure, que es poden aconseguir a Internet.

- Com a sistema gestor de bases de dades SGBD, **Oracle 10g Express Edition**, es pot trobar a <http://www.oracle.com/technetwork/database/express-edition/downloads/102xewinsoft-090667.html> i ha estat lliurat amb el material de l'assignatura.
- Entorn gràfic per al desenvolupament de les bases de dades **SQL Developer**, lliurat amb el material de l'assignatura i disponible a <http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html>
- Per a la redacció de tots els documents escrits, com aquest pla de treball o la memòria, i per a la presentació virtual, s'ha utilitzat la suite OpenOffice 3. Aquesta suite, ara propietat d'Oracle, es pot trobar a <http://www.softcatala.org/wiki/Rebost:OpenOffice.org>. Les diferents opcions que ofereix el processador de textos, com control de canvis, navegació per diferents parts del document utilitzant els encapçalaments o la creació automàtica de la taula de continguts han facilitat molt la feina. També és de valorar que, a diferència de versions anteriors d'OpenOffice, a la versió utilitzada la funció de conversió a pdf genera el document amb marcadors de navegació d'acord amb els nivells d'esquema del document original.
- No s'ha utilitzat l'eina de gestió de projectes OpenProj, com es va posar a la planificació inicial. Per als cronogrames i diagrames de Gantt ha estat suficient amb els gràfics del full de càlcul de la suite OpenOffice.
- Com a ajuda per a l'elaboració de tots els scripts s'ha utilitzat l'editor de text **Context**. És gratuït i d'utilització molt senzilla i intuïtiva i es pot aconseguir de la pàgina <http://www.contexteditor.com>. Entre altres funcions ofereix ressaltat de text compatible amb diferents llenguatges, entre ells SQL.
- Per al disseny de la base de dades s'ha optat pel programa DB Designer 4, que es pot trobar a <http://fabforce.net/dbdesigner4/index.php>. Altres candidats en el moment de fer la planificació inicial eren el Diagram Designer de Meesoft, de FabForce o el MagicDraw.

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

#### Anàlisi de requisits

El segon capítol es dedicarà a estudiar quines són les necessitats reals del sistema.

#### Scripts de la BD

En aquest capítol s'inclouran les informacions relatives a la creació de scripts, en tres apartats:

- Scripts de creació de taules.
- Scripts per a la creació de procediments i disparadors.
- Scripts d'inserció de registres (els necessaris per arrencar i els de prova).

## Presentació virtual

## Conclusions

### 2 Planificació del projecte.

La planificació del projecte comença per l'elaboració d'un pla de treball, ja lliurat, on hi consten les diferents feines pendents per fer i una proposta de temporització per a cada una de les feines projectades. A mesura que ha avançat la feina s'ha vist com era convenient fer alguns canvis en aquesta planificació.

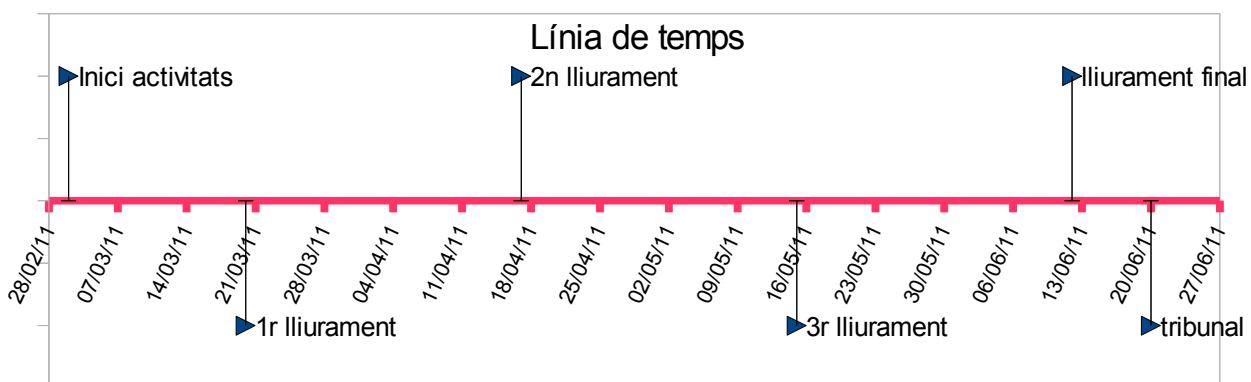
#### 2.1 Planificació inicial del projecte

La planificació inicial va ser el primer lliurament fet. Aquesta planificació inicial no era completa, i es va detallant més a la memòria.

#### Temporització

Per a calcular el temps necessari per a cada una de les tasques és necessari tenir en compte les dates per a lliuraments fixades per l'assignatura. Aquestes dates estan fixades al pla docent:

Event	Inici activitats	1r lliurament	2n lliurament	3r lliurament	lliurament final	tribunal
data	02/03/11	20/03/11	17/04/11	15/05/11	12/06/11	20/06/11

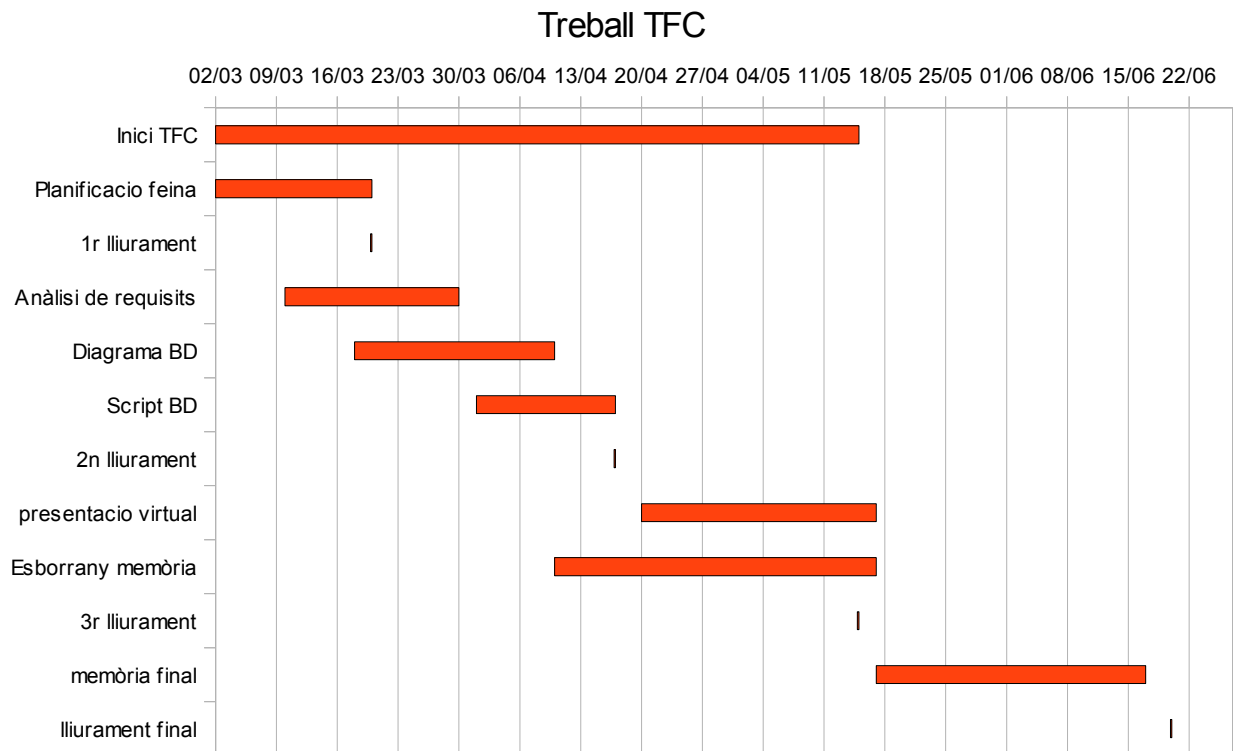


La distribució inicial de les tasques es va fer adaptant-se a aquesta temporització, marcant quines tasques s'inclourien a cada un dels lliuraments.

La primera planificació va ser:

1r lliurament	Pla de treball
2n lliurament	Anàlisi de requisits de la BD Diagrama de la BD Scripts de la BD (taules i vistes)
3r lliurament	Memòria Presentació virtual del projecte
Lliurament final	Documents revisats: -Diagrama BD -Scripts (amb disparadors i procediments) -Memòria -Presentació

D'aquesta manera la distribució temporal de les tasques es va representar mitjançant el següent diagrama de Gantt:



## 2.2 Modificacions de la planificació

La planificació inicial s'ha anat modificant en diferents aspectes:

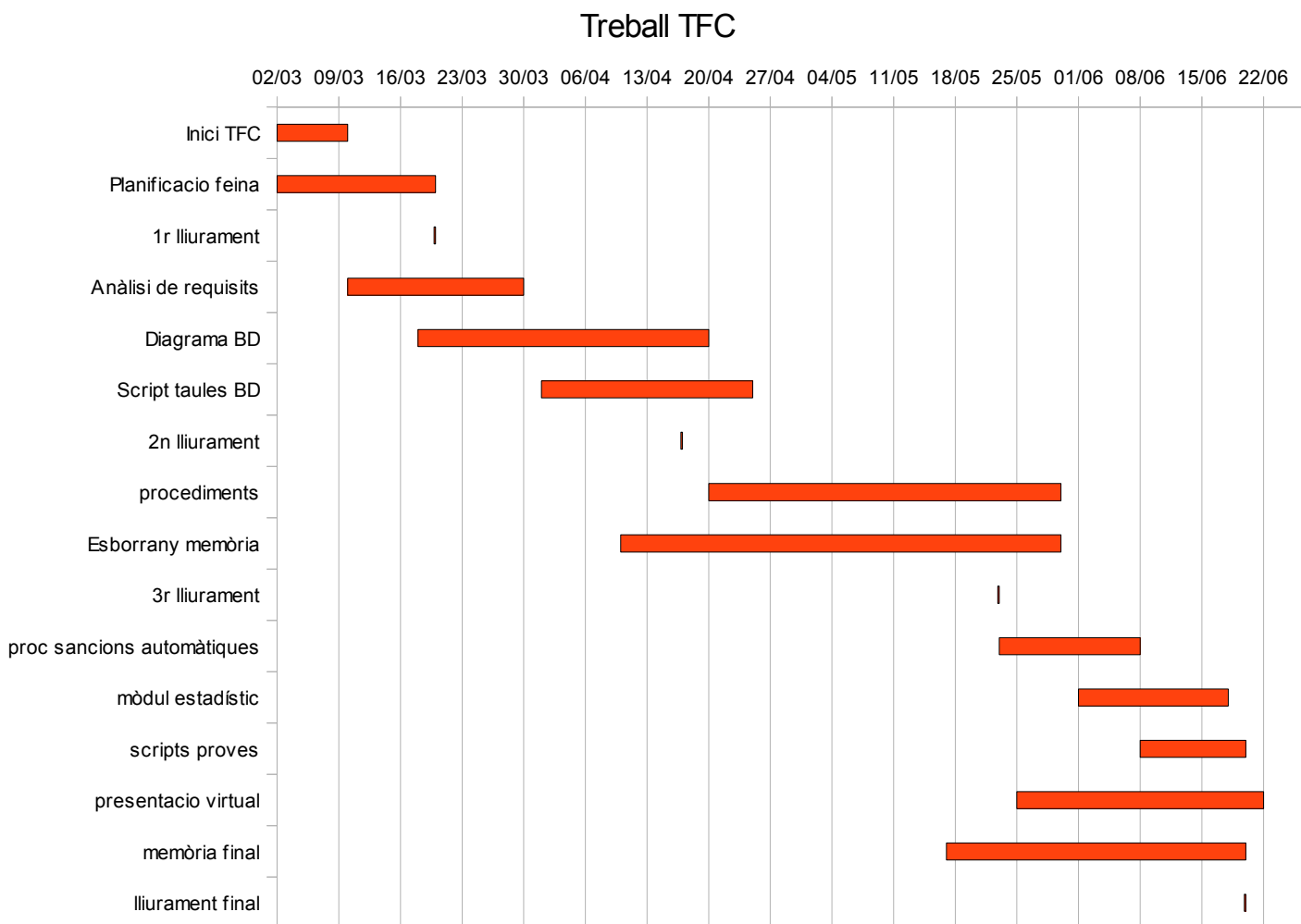
- S'ha considerat oportú deixar la presentació virtual per al darrer lliurament, i centrar la feina del tercer lliurament en l'elaboració dels procediments emmagatzemats.
- La memòria es va redactant al llarg de tot el projecte, sempre ampliant la part que correspon a la feina que s'està fent en aquell moment. Això facilita la seva redacció, ja que es van fer les anotacions en el moment que es prenen les decisions, sense haver de fer un posterior



esforç addicional per recordar-les. D'altra banda serveix per a dur un seguiment de la pròpia feina per a fases posteriors. A la fase de preparació de procediments, per exemple, es van consultar les anotacions fetes en l'apartat de disseny de taules. D'aquesta manera, a partir del 2n lliurament es va lliurant cada vegada una còpia de la memòria en l'estat que es troba.

- Al tercer lliurament es varen revisar els dissenys d'algunes taules, per poder satisfer els requisits demanats a l'enunciat. En particular es varen crear la taula per al log dels procediments, i es varen redissenyar les taules que conformen el mòdul estadístic i les taules necessàries per a la generació automàtica de sancions.

Amb els canvis, la planificació corregida per al tercer lliurament va quedar de la forma següent:



### 3 Anàlisi dels requisits.

Al plantejament inicial cal tenir en compte el fet que per normativa legal totes les actuacions disciplinàries han de prescriure a final de curs. Això implica que a començament de curs tots els alumnes han de ser tractats per igual, partint amb zero amonestacions i sancions. El mòdul estadístic que contempla sancions d'anys anteriors només es podrà utilitzar a nivell d'estudi per analitzar l'efecte de accions disciplinàries noves o l'evolució dels alumnes segons el tipus de sanció que se'ls va aplicar en cursos anteriors, per exemple. Però les dades de sancions i amonestacions de cursos anteriors no haurien de ser visibles per als usuaris habituals de la BD.

Els requisits s'extreuen bàsicament del llistat de 15 requeriments que s'esmenten a l'enunciat, i tot el disseny de la BD es farà de forma que es puguin aconseguir. Els requeriments de l'enunciat són els següents:

1. El model de dades ha de permetre introduir i mantenir les dades bàsiques del alumnes i professors del centre corresponent.
2. El model de dades ha de permetre introduir i mantenir els diferents cursos que s'imparteixin anualment i les assignatures de cada un d'ells.
3. Caldrà donar la possibilitat d'introduir els diferents calendaris escolars per a cada curs definit. Com a dades fonamentals, el calendari ha de permetre introduir el curs, l'assignatura i l'horari d'aquesta assignatura (dia de la setmana i hora). Aquest calendari ha de contemplar els dies festius definits per la conselleria d'educació.
4. El model de dades definit ha de permetre definir amonestacions amb les dades bàsiques següents: alumne amonestat, data, hora, curs, assignatura, professor responsable, tipus, gravetat, i si s'ha comunicat als pares. Es valorarà el fet d'afegir noves dades que serveixin per donar més valor a la gestió d'aquestes amonestacions.
5. Existiran uns tipus d'amonestacions establertes pels professors.
6. El model de dades ha de permetre definir noves amonestacions fàcilment, és a dir, sense necessitat de canviar l'estructura de les taules que les emmagatzemaran o modificar els procediments emmagatzemats definits.
7. En cas de tenir una amonestació prou greu o una acumulació d'amonestacions s'ha de poder introduir una sanció indicant els motius i la resolució de la mateixa.
8. El sistema ha de permetre suportar el règim de sancions que defineixi cada un dels centres educatius. Com a element bàsic, cada sanció ha d'anar lligada a un tipus d'amonestació definit.
9. Existiran uns tipus de sancions establerts pels professors.
10. La definició de les regles per activar sancions automàtiques s'ha d'emmagatzemar a base de dades i s'ha de permetre definir noves regles o modificar les existents fàcilment, és a dir, sense modificar l'estructura de taules o de les sancions emmagatzemades.
11. El model de dades ha de permetre saber a quin curs pertany cada alumne, quins professors donen cada assignatura i quins professors són responsables de cada curs.
12. S'han d'emmagatzemar els dies i les hores d'atenció als pares per part dels professors així com les hores de consulta als alumnes.
13. L'aplicació ha de permetre conèixer, en qualsevol moment, quines amonestacions i sancions

se li han assignat a cada alumne.

14. L'aplicació haurà de disposar, com a mínim, de les funcionalitats següents tot complint amb els requisits expressats prèviament:

- Procediments d'ABM (Alta + Baixa + Modificació) dels alumnes
- Procediments d'ABM dels professors
- Procediments d'ABM dels cursos
- Procediments d'ABM de les assignatures
- Procediments d'ABM del calendari escolar
- Procediments d'ABM de les diferents amonestacions i sancions incloent la possibilitat de definir-ne noves tipologies
- Procediments de consulta de que permetin obtenir:
  - a. Llistat de totes les amonestacions imposades indicant-ne la seva informació bàsica.
  - b. Llistat de tots els Alumnes d'un curs indicant-ne la seva informació bàsica.
  - c. Llistat de tots els tipus d'amonestacions i sancions disponibles per a aplicar en el centre.
  - d. Llistat de totes les amonestacions i sancions d'un alumne. • Es valorarà la implementació d'altres procediments o funcionalitats que puguin ser d'utilitat, sempre i quan estiguin documentades i consensuades amb el consultor.

15. Mòdul estadístic, una part molt IMPORTANT del treball és la implementació d'un mòdul estadístic que s'ha d'alimentar a partir dels procediments que implementin les funcionalitats esmentades, per tal d'oferir les dades següents en temps constant 1, és a dir, fer una SELECT sobre un registre d'una taula (que no sigui una vista calculada o materialitzada, ni fent servir funcions d'agregats : sum, avg, etc .. amb group by).

El mòdul estadístic haurà de donar resposta a les consultes següents:

1. El número d'amonestacions per alumne (independentment del curs)
2. El número de sancions per alumne i any.
3. La mitjana d'amonestacions per professor i any.
4. El número de sancions per curs i any.
5. El nom de l'alumne més sancionat en un any donat.
6. El nom del professor més amonestador (que posa més amonestacions) per curs.
7. La mitjana de sancions que tenen els alumnes per curs.
8. El numero d'alumnes que no tenen cap amonestació.

### 3.1 Muntatge de la BD

Per al muntatge de la BD és necessari crear els diferents elements que la formen en l'ordre adequat. L'ordre a seguir és el següent:

1. Taules i vistes (*disciplinaDDL.sql*)
2. Disparadors (*disciplinaTRIGGER.sql*)
3. Funcions (*disciplinaFUNC.sql*)
4. Procediments (*disciplinaPROC.sql*)
5. Introducció de registres permanents (*disciplinaDML.sql*)
6. Introducció de registres de prova (*DADES\_EXEMPLE.sql*)
7. També es poden introduir registres des de l'arxiu de joc de proves per als procediments, *provesPROCEDIMENTS.sql*, que es comenten al capítol 6 de la memòria.

## 4 Disseny de la BD

Com a eina per al desenvolupament gràfic del disseny s'ha optat finalment per utilitzar el programa DBDesigner 4 de FabForce. El motiu d'aquesta decisió és que es tracta d'un programa gratuït bastant complet, i prou còmode per fer el disseny d'aquesta base de dades. S'adjunta a la memòria un arxiu amb el disseny de la BD en format gràfic per a millorar la seva visualització, tot i què ve inserit dins aquest document a l'apartat dels annexos.

### 4.1 Taules

A continuació hi ha una descripció de les taules del projecte, amb la justificació d'algunes decisions preses quant al seu disseny. Algunes de les taules, com *tipus\_sancio*, *regles\_sancio* o *log\_proc* s'han afegit al tercer lliurament, no eren al disseny inicial de la BD.

#### Curs\_escolar

Aquesta taula serà la referència per moltes de les altres taules, donat que per la majoria de funcions la base de dades ha de funcionar com si les úniques dades fossin les del present curs. Les consultes a dades d'anys anteriors es podran fer amb finalitats estadístiques, i sense modificar les dades. Simplement s'emmagatzemen el nom del curs escolar en format de text ("2011-2012", per exemple) i les dates inicial i final que corresponen respectivament al primer i darrer dia lectiu del curs. Hi ha la restricció de què la data inicial ha de ser anterior a la final.

#### Festius

Per a simplificar el requisit de demanar el calendari del curs escolar es guarden únicament els dies festius i períodes de vacances que hi ha dins cada curs. A cada període de vacances se li assigna també un nom, com podria ser "Vacances de nadal" o "Festivitat de Sant Antoni". Els períodes de vacances poden ser d'un o més dies, i tant si són definits per la Conselleria d'Educació, festes locals o de lliure designació del centre tendran la mateixa consideració a nivell de la base de dades. En cas de què el camp *data\_final* tenguí el valor null s'interpretarà que es tracta d'un únic dia festiu, no d'un període de vacances.

Posteriorment es crearà un procediment emmagatzemat per a calcular si una data en concret és vàlida, on s'haurà de comprovar:

- La data està entre les dates inicial i final de la taula *curs\_escolar*
- La data no es troba dins cap dels períodes festius especificats per al curs escolar

- La data no correspon a cap de setmana

A nivell de taules de la base de dades, el tractament de calendari i d'horari setmanal es farà de forma separada.

## Nivells

Aquesta taula emmagatzema el nom dels diferents nivells educatius que s'imparteixen al centre. Serveix de referència per altres taules, com ara els grups.

## Grups

El nombre de grups creats per a cada nivell pot anar variant d'un any a un altre, ja que depèn del nombre d'alumnes que es preveu que es matriculi. Una vegada creats el curs escolar corresponent es podran crear tots els grups. Una vegada matriculats els alumnes i donats d'alta els professors es podrà indicar qui són el tutor del curs (professor) i el delegat (alumne matriculat a aquest grup).

## Departaments

Als centres de secundària els professors s'organitzen en departaments didàctics. Els professors han de pertànyer a un dels departaments del centre.

## Professors

A l'igual que a la taula anterior, aquí van les dades que no canvien referents al professorat. El camp clau és el NIF de la persona. Els camps que no poden ser nuls són el NIF, nom i llinatge1.

Per saber si un professor de la taula està donat d'alta per a un curs escolar s'ha de comprovar si se l'ha assignat per impartir alguna matèria. Això es pot comprovar fent una consulta a la taula *submateries*.

## Horari\_atencio

En aquesta taula s'emmagatzema la informació d'horari d'atenció de cada professor per a cada curs escolar. L'horari és d'una hora setmanal tant per pares com per alumnes, i es codifica com un nombre entre 1 i 5 pel dia entre dilluns i divendres i un nombre entre 1 i 7 per a la hora, com es descriu a l'apartat de la taula *horaris\_submateria*.

## Rols

Emmagatzema els possibles rols que poden tenir els usuaris de la taula anterior, per especificar quins drets té cada un d'ells.

## Usuaris

No tots els professors tenen perquè accedir directament a la base de dades. Correspondrà a l'administrador del sistema crear els usuaris que es creguin oportuns, amb els permisos corresponents. Cada usuari estarà relacionat únicament amb una persona, i pot ser un professor o un administratiu del centre.

Els procediments de la base de dades no contempen distinció entre diferents usuaris, sino que

hi ha un únic usuari definit per accedir a la BD. Els drets atorgats a cada tipus d'usuari es poden tenir en compte en el sistema de validació d'una futura aplicació programada en llenguatge d'alt nivell. Per exemple, en la creació de menús de l'aplicació segons el rol assignat.

## Directius

Els membres de l'equip directiu són els que tendran més privilegis en l'accés a la BD. Són els que tenen dret a imposar sancions i els que tenen accés a dades d'anys anteriors. Els membres de l'equip directiu també són professors del centre, per tant les seves dades personals i de contacte s'emmagatzemaran a la taula professors.

Els càrrecs directius també poden variar d'un curs a un altre, per això s'ha d'especificar quin professor ocupa cada càrrec a cada curs escolar.

## Càrrecs

Els diferents càrrecs que formen l'equip directiu, taula referenciada per la taula *directius*.

## Alumnes

Aquesta taula conté la informació sobre els alumnes, que es conserva d'un curs a un altre si no es modifica expressament. Hi ha les dades personals, de contacte, i el número d'expedient. Les dades que canvien d'un curs a un altre, com pot ser el grup al qual pertanyen, no es guarda aquí. El camp clau és el nombre d'expedient. És un camp numèric, tipus integer, i no s'ha definit com a autonumèric perquè es preveu que tots els números d'expedient vinguin donats pel programa de gestió de la Conselleria d'Educació.

## Alumnes\_matrícula

Per a cada alumne es guardarà la informació de a quin nivell i grup estava matriculat per a cada curs acadèmic. Totes les actuacions disciplinària referent a un alumne s'han de emmarcar en un dels cursos escolars en què l'alumne estava matriculat.

S'ha substituït el camp *n\_amonestacions* inclòs al disseny inicial per un camp a la taula *amonestacions\_alumne* que permet a més de controlar si les amonestacions han estat comptabilitzades fer una distinció per tipus d'amonestació.

## Materies

Fa referència a les diferents matèries que hi pot haver als horaris, per nivell. Es definiran a partir del pla d'estudis. Per a cada nivell s'estableix quines matèries s'han d'impartir, donant el codi i una descripció. Els camps són (*nivell, codi, nom*) i la clau principal és (*nivell, codi*).

## Submateries

Per a cada curs escolar, són els grups de cada matèria que realment es creen al centre. De matemàtiques de 3r d'ESO, per exemple, hi pot haver matemàtiques a 3rA, 3rB i 3rC per al curs 2010-11, i podrien ser 3rA, 3rB, 3rC i 3rD per al 2011-12. Per tant per a cada submatèria s'ha d'especificar el curs escolar al qual pertany i la matèria (donada per nivell i codi) a la que correspon.

Cada submatèria estarà assignada a un professor del centre. Consideram que un professor està

donat d'alta en un curs escolar concret si està assignat a alguna de les submatèries del curs.

Els camps d'aquesta taula són (*curs\_escolar, nivell, codi, submateria, professors\_nif*), i la clau principal (*curs\_escolar, nivell, codi, submateria*).

El concepte de submatèria apareix al programa de gestió docent *Gestib*, d'obligada utilització per a tots els centres públics de primària i secundària de la Comunitat Autònoma de les Illes Balears. S'ha optat per utilitzar aquest concepte aquí per a una possible compatibilitat de dades amb aquest programa.

## Submateries\_grup

Aquesta taula té la funció d'establir per a cada curs acadèmic la relació m:n entre els cursos que hi ha i les submatèries que s'imparteixen. En el cas de matèries optatives, per exemple, la matèria Processos de Comunicació de 3r pot tenir la submatèria PCOM\_ABC i que aquesta s'imparteixi a alumnes de 3rA, 3rB i 3rC. S'ha d'especificar quina submatèria i quin grup es relacionen a cada entrada, per tant els camps són (*curs\_escolar, nivell, codi, submateria, grup*), i tots formen part de la clau principal.

## Horaris\_submateria

Per a cada submateria (que s'identifica per *curs\_escolar, nivell, matèria* i el nom de la submateria) s'introduiran els períodes setmanals en què s'imparteixen indicant el dia de la setmana (nombre enter entre 1 i 5 per indicar de dilluns a divendres) i hora lectiva (nombre entre 1 i 7 per indicar entre la primera i darrera hora lectiva del dia). Així, 2,3 indicaria dimarts a 3a hora i 4,6 indicaria dijous a 6a. Aquesta notació és la que utilitzen els programes generadors d'horaris GP-Untis i Kronowin, dos dels més utilitzats a centres educatius.

Els camps que formen la taula són (*curs\_escolar, nivell, codi, submateria, dia, hora*), i tots formen part de la clau principal. Per a cada submatèria hi haurà tantes entrades en aquesta taula com classes setmanals s'imparteixin.

## Amonestacions

En aquesta taula es guardaran totes les amonestacions posades al centre. Cada amonestació es posada per un únic professor, i entre els diferents camps de la taula hi ha el tipus d'amonestació vinculat a la taula corresponent, un camp per observacions que permet fer una descripció més detallada dels fets, i la data i hora de l'amonestació. El camp clau és (*id, curs\_academic*), on *id* és un camp numèric que comença una nova numeració per a cada curs.

## Tipus\_amonestacio

Els tipus més comuns de sanció es guarden en aquesta taula. El fet de tenir tipificades les amonestacions agilitza el fet de posar-ne una i facilita les estadístiques posteriors.

## Amonestacions\_alumne

Estableix la relació entre les taules alumne i amonestacions, per permetre que una mateixa amonestació es pugui posar a més d'un alumne. Això permet posteriorment veure que un grup d'alumnes estaven involucrats en els mateixos fets, cosa que seria més difícil si s'hagués de crear una amonestació individual per a cada un dels alumnes. Si una amonestació per barallar-se se posa a

dos alumnes, es veu que eren dins la mateixa baralla, cosa que no es pot detectar si dos alumnes tenen una amonestació per barallar-se encara que la data, hora i descripció de l'amonestació sigui la mateixa.

De tota manera s'afegeix el camp observacions per si a l'amonestació s'ha de fer alguna distinció entre un alumne i un altre.

La data de comunicació als pares i la informació de si s'ha hagut de treure l'alumne de classe s'emmagatzema en aquesta taula enlloc d'amonestacions, ja que podria ser una informació diferent per a cada alumne implicat.

S'ha afegit un camp per indicar si la sanció ha estat comptabilitzada per a la generació de sancions, siguin manuals o automàtiques. Quan es crea una sanció les amonestacions que l'han provocat queden "marcades", per no ser tingudes en compte per a sancions posteriors.

## Sancions

Les sancions comporten un càstig imposat a un alumne. Les sancions poden ser originades per fets greus, o per acumulació d'amonestacions. En aquest darrer cas les sancions seran generades automàticament segons les regles de generació de sancions que s'estableixen a la taula *regles\_sancio*. Les sancions automàtiques hauran de ser validades per un cap d'estudis per ser efectives. Les sancions que tinguin *null* el camp corresponent a càrrec es consideraran pendents d'aprovació, tant si són generades manualment com de forma automàtica.

En els casos en què els fets hagin ocasionat danys materials al centre o a altres persones, hi cap la possibilitat de reclamar que la família es faci càrrec de part o la totalitat de les despeses ocasionades. Aquestes quantitats es poden satisfer en diferents pagaments. Per aquests casos hi ha els camps *quantia* i *pendent*, per emmagatzemar respectivament la quantitat total a pagar i la part que encara queda per pagar.

## Tipus\_sancio

S'emmagatzemen els tipus preestablerts de sanció. Aquesta classificació de les sancions pot facilitar el posterior tractament estadístic de la base de dades, i a més possibilita la implementació de sancions generades automàticament per acumulació d'un tipus concret d'amonestació.

Les sancions manuals no se generen de forma automàtica, els altres tipus de sanció es podran generar automàticament si hi ha creada la regla de creació corresponent. Les sancions manuals o personalitzades són les marcades amb el tipus 'pers'.

## Regles\_sancio

Aquesta taula guarda la informació necessària per poder generar de forma automàtica un tipus de sanció. El fet de guardar les regles a una taula i no directament als procediments de la BD o a un posterior programa d'alt nivell facilita que els futurs usuaris puguin personalitzar la creació de sancions a cada centre.

En un principi s'havia posat com a camp clau el tipus de sanció. Això es va descartar perquè només permetia una única regla de generació de sancions automàtiques per a cada tipus.

## Pagaments

S'indica la quantitat que s'ha pagat d'una sanció, i el dia en què s'ha fet el pagament. Si una



sanció s'ha pagat en diferents dies, la suma de les quantitats pagades no ha d'excedir la quantia total de la sanció. En el moment d'afegir un pagament a la taula s'ha d'actualitzar el camp *pendent* de la sanció corresponent a la taula *sancions*.

## Log\_proc

La taula *log\_proc* s'utilitzarà per a guardar les entrades corresponents a les crides fetes a procediments que s'hagin executat de forma satisfactòria. Es guardarà la data d'execució, el nom del procediment, els paràmetres d'entrada i els paràmetres de sortida.

## 4.2 Vistes

S'han creat diferents vistes per a simplificar el posterior disseny d'alguns mètodes i disparadors. Principalment s'han dissenyat per a relacionar taules que tenen camps separats i què en algunes consultes s'hauran de comprovar conjuntament.

### Amonestacions professor

A la taula *amonestacions\_alumne*, que és la que té la informació de quantes amonestacions s'han posat realment, no surt informació de quin professor ha posat l'amonestació ni de a quin nivell i grup està matriculat l'alumne.

A la vista *amonestacions\_professor* s'ha posat per a cada amonestació posada a un alumne la informació de a quin curs escolar, professor, num amonestació, expedient d'alumne, nivell i grup correspon.

### Llistat grups

Aquesta vista és útil per treure llistats de grups, on es vegin dades de matrícula com curs escolar, nivell o grup a un mateix llistat que dades personals com nom, llinatges, adreça o telèfon.

### Dir

Aquesta vista mostra el llistat de càrrecs directius de cada curs escolar, mostrant nom i llinatges.

## 4.3 Taules del mòdul estadístic

El mòdul estadístic requereix un tractament diferent a la resta de taules. Està dissenyat per donar resposta al Mantindrà actualitzada la informació necessària per poder demanar dades estadístiques en una simple instrucció SELECT, tal i com es demana a l'enunciat, concretament al punt 15 dels requisits funcionals del sistema.

Utilitzant disparadors, els procediments que actualitzin les taules es posaran en funcionament quan canviï la informació rellevant per a les estadístiques: quan es posi una nova sanció canviarà el nombre de sancions posades i s'actualitzaran els camps corresponents, tant a nivell global per al curs escolar com per al nivell concret al qual pertany l'alumne sancionat.; es farà la comprovació de si hi ha canvis a la informació d'alumne més sancionat. El mateix es farà quan s'afegixi una amonestació, amb el nombre total d'amonestacions i professor més amonestador.

La informacions de número de professors i d'alumnes totals i per nivell s'haurà d'actualitzar quan es donin altes o baixes de professors o d'alumnes.

Hi ha d'haver procediments per inicialitzar les taules, que serviran per calcular els valors en cas d'importació de dades des de fonts externes o en cas de corrupció de la base de dades. Per al funcionament normal de la base de dades amb disparadors únicament s'actualitzaran els valors afectats.

## Estadístiques

La taula *estadistiques* emmagatzemarà informacions de caire global per a cada curs escolar. serà accessible per a l'equip directiu. No s'hi faran insercions directament, sinó que els camps s'aniran actualitzant automàticament quan es modifiquin altres taules, utilitzant disparadors. Constarà de diferents columnes, la primera de les quals serà camp clau i correspondrà al curs escolar. La resta de camps contendrà els valors necessaris per calcular les estadístiques demanades a l'enunciat. Aquest requisit especifica que les consultes s'han de fer directament a una taula, sense utilitzar funcions agregades. Per tant, una solució és tenir la informació necessària per a les estadístiques que es demanen a una taula, i anar actualitzant el seu valor cada vegada que es fa un canvi.

En concret, les informacions que s'han de tenir emmagatzemades són les següents:

- Nombre total d'alumnes.
- Nombre total de professors.
- Nombre total d'amonestacions.
- Nombre total de sancions.
- Nombre d'alumnes amb alguna amonestació.
- Nombre d'alumnes amb alguna sanció.
- Nombre de professors que han posat alguna amonestació.

Els valors d'aquesta taula es van actualitzant mitjançant disparadors que s'activen en el moment que s'insereixen registres nous a les taules que els afecten. Així, si tenim emmagatzemat el nombre d'amonestacions posades, quan s'afegeixi una nova amonestació a un alumne bastarà incrementar en un el valor del nombre total d'amonestacions de la taula estadístiques i per seguir tenint la taula actualitzada, sense haver de fer una consulta a tota la base de dades per recomptar el nombre d'amonestacions. Després caldria comprovar el nombre d'amonestacions tant de l'alumne que l'ha rebut com del professor que l'ha posat, per si hi hagués canvis en el professor o alumne amb més amonestacions.

## Estadístiques\_nivell

Aquesta taula complementa l'anterior, i conté les dades corresponents a cada nivell educatiu. Es fa d'aquesta manera ja que els nivells educatius que té el centre es troben emmagatzemats a una taula, i es poden modificar. Per a cada curs escolar, aquesta taula tindrà tantes files com nivells, i per a cada nivell es guardarà la informació de

- Nombre d'alumnes.

- Nombre de professors.
- Nombre d'amonestacions.
- Nombre de sancions.
- Nombre d'alumnes amb alguna amonestació.
- Nombre d'alumnes amb alguna sanció.
- Nombre de professors que han posat alguna amonestació.

### Alumne més sancionat

La informació d'alumne més sancionat es guardarà a la taula *alumne\_mes\_sancionat*, i podrà tenir més d'una entrada per cada curs acadèmic i nivell. En cas de què un alumne que no era a la taula arribi al màxim de sancions seria afegit. En cas que un alumne superàs la resta d'alumnes empatats, els que han quedat superats serien eliminats de la taula.

Els alumnes amb major nombre de sancions per a un curs sense separar per nivells s'especificarà deixant *null* el camp corresponent a *nivell*.

Els camps que formen aquesta taula són

*CURS\_ESCOLAR, NIVELL, N\_EXP, N\_SANCIONS*

### Alumne més amonestat

La informació d'alumne més amonestat es guardarà a la taula *alumne\_mes\_amonestat*, que tindrà un funcionament anàleg al de la taula per l'alumne més sancionat. Els alumnes amb més amonestacions per a un nivell educatiu s'especificaran posant el nivell al qual pertanyen al camp *nivell*. Els que tenguin el màxim d'amonestacions a nivell de tot el centre s'especificaran posant *null* al camp *nivell*. Els camp que formen aquesta taula són

*CURS\_ESCOLAR, NIVELL, N\_EXP, N\_AMONESTACIONS*

### Professor més amonestador

El funcionament de la taula *professor\_mes\_amonestador* serà semblant al de la taula *alumne\_mes\_sancionat*, utilitzant dades de professors i amonestacions enlloc d'alumnes i sancions. Per a cada curs s'especificarà el professor més amonestador globalment, i el que té major nombre d'amonestacions per a cada nivell. Els camps que formen la taula són

*CURS\_ESCOLAR, NIVELL, NIF, N\_AMONESTACIONS*

### Estadístiques\_alumne

Per evitar haver de calcular totals en els càlculs estadístics, sigui en el moment de fer consultes o en el moment d'actualitzar taules, s'ha optat per crear taules per a emmagatzemar el nombre d'amonestacions i sancions que té cada alumne a cada curs escolar. Aquesta taula s'haurà de mantenir de forma transparent a l'usuari mitjançant disparadors. Es crearà una nova entrada a la taula cada vegada que un alumne es matriculi, i el valor de les seves amonestacions i sancions s'anirà incrementant i decremantant segons es vagin creant o eliminant amonestacions o sancions.

D'aquesta manera, la taula tindrà tantes entrades per a un curs escolar com alumnes

matriculats tingui. Els camps d'aquesta taula són

*CURS\_ESCOLAR, NIVELL, N\_EXP, N\_AMONESTACIONS, N\_SANCIONS*

### Estadístiques\_professor

De manera semblant a la descrita per a la taula anterior, es crearà una entrada per a cada nivell i professor que imparteixi classes. Es crearà entrada per a un curs escolar, nivell i professor sempre que hi hagi alguna submatèria que vinculi el professor amb el nivell i curs. La creació de entrades a aquesta taula també es farà mitjançant disparadors.

Els camps d'aquesta taula seran

*CURS\_ESCOLAR, NIF, NIVELL, N\_AMONESTACIONS*

Per a un curs escolar un professor tindrà tantes entrades com nivells de classe imparteixi, a més d'una fila amb el valor *nivell=null* que guardarà el nombre de sancions posades a nivell de tot el centre.

## 4.4 Vistes del mòdul estadístic

A les taules descrites anteriorment es guarda la informació necessària per a calcular les estadístiques demanades a l'enunciat. Una vegada s'aconsegueix que les taules estiguin actualitzades és senzill aconseguir els valors demanats. Això es fa mitjançant les vistes que es descriuran a continuació.

### calculs\_estadistics

Aquesta vista recull diferents càlculs que fan referència a valors del curs escolar, sense fer distinció entre nivells educatius. Les diferents columnes que hi apareixen són:

*curs\_escolar*

*amo\_alum*: nombre d'amonestacions per alumne

*san\_alum*: nombre de sancions per alumne

*amo\_prof*: nombre d'amonestacions per professor

*sense\_amo*: nombre d'alumnes sense amonestació

*sense\_percent*: percentatge sobre el total dels alumnes sense amonestació

### calculs\_estadistics\_nivell

Aquesta vista consta d'exactament els mateixos paràmetres que la consulta anterior, però separant els resultats per nivell educatiu. Les dues primeres columnes són *curs\_escolar* i *nivell*, i la resta de columnes coincideixen amb les descrites abans.

### Ranking\_amo\_alumnes

Aquesta vista és equivalent a la taula *alumne\_mes\_amonestat*, amb la diferència que ara apareixen nom i llinatge enlloc el número d'expedient. Apareixen els noms i llinatges dels alumnes i el nombre d'amonestació dels alumnes amb el nombre més alt d'amonestacions per nivell, i els que a més tenen el nombre més alt d'amonestacions de tot el centre apareixen una vegada per al seu nivell

i una altra per al rànquing del centre, amb el valor null per al camp *nivell*.

### Ranking\_san\_alumnes

Com a la taula anterior, però comptabilitzant sancions enlloc d'amonestacions.

### Ranking\_amo\_profs

Presenta nom, llinatges i nombre d'amonestacions posades dels professors més amonestadors del centre, i també per a cada nivell.

## 5 Scripts de la BD

### 5.1 Taules

Els scripts de creació de taules s'han elaborat amb l'ajuda del programa de disseny DBDesigner4, ja que permet generar codi SQL directament una vegada creada la definició de les taules. El codi generat per aquest programa, però, ha hagut de ser retocat per adaptar-se a la sintaxi concreta de l'Oracle.

Els scripts de creació es presenten com a annex al document *disciplinaDDL.sql*. i els scripts d'introducció de dades de prova al document *disciplinaDML.sql*. Aixímateix s'inclou l'script *disciplinaDROP.sql*, que s'ha utilitzat per eliminar totes les taules a mesura que s'anava provant la seva execució.

### 5.2 Consultes

#### Llistat\_grups

Treu un llistat amb nom i llinatges dels alumnes de cada grup, ordenat per curs escolar, nivell, grup, llinatges i nom. Agafa les dades de les taules *alumnes* i *alumnes\_matricula*.

### 5.3 Funcions

Per facilitar el disseny de l'aplicació, s'ha començat per dissenyar funcions que seran utilitzades des de diferents procediments. Són en major part funcions que retornen un valor booleà i que serveixen per fer comprovacions comuns a diferents procediments.

#### Bool\_to\_string

Aquesta és una funció que no forma part de les funcionalitats de la BD pròpiament, sinó que es tracta d'una funció auxiliar per facilitar la depuració de les diferents funcions. Retorna la cadena 'TRUE' o 'FALSE' segons el valor del paràmetre booleà d'entrada, facilitant la visualització de la sortida de la resta de funcions.

## Curs vàlid

Comprova si el curs passat com a argument existeix a la base de dades. Retorna vertader si és així, fals en cas contrari. La signatura és

```
curs_valid (curs_escolar_ IN VARCHAR2, RSP OUT STRING): BOOLEAN
```

on curs\_escolar\_ és el curs que es vol comprovar, RSP és la cadena per missatge de resposta amb valor 'OK' o 'ERROR ' si s'ha generat una excepció.

## Curs té nivell

Comprova si hi ha algun grup del nivell passat com a paràmetre creat per al curs escolar. La signatura és

```
CURS_TE_NIVELL(curs_escolar_ VARCHAR2, nivell_ VARCHAR2, RSP OUT STRING)
```

En concret, no és possible matricular un alumne a un nivell on no hi ha grups creats.

## Alumne vàlid

Comprova si un número d'expedient existeix a la taula alumnes. La signatura és

```
alumne_valid(exp_ IN INTEGER, RSP OUT STRING):BOOLEAN
```

on exp\_ és el número d'expedient que es vol comprovar, RSP és la cadena per missatge de resposta amb valor 'OK' o 'ERROR ' si s'ha generat una excepció.

## Alumne matriculat curs

Comprova si un número d'expedient existeix i si correspon a un alumne matriculat al curs passat també com a paràmetre. Prèviament es criden les funcions *curs\_valid* i *alumne\_valid* per generar excepcions en cas que algun dels paràmetres no existeixi a la base de dades. La signatura és

```
alumne_matricula_curs(exp_ IN INTEGER, (curs_escolar_ IN VARCHAR2,RSP OUT STRING):BOOLEAN
```

on exp\_ és el número d'expedient que es vol comprovar, curs\_escolar\_ és el curs escolar en el que volem saber si l'alumne està matriculat, RSP és la cadena per missatge de resposta amb valor 'OK' o 'ERROR ' si s'ha generat una excepció.

## Data vàlida

Comprova si una data correspon a un dia lectiu dins un curs escolar. Per a cada curs escolar hi ha una data inicial i una data final. Per comprovar si una data és vàlida es fan les següents comprovacions:

- Que les dates del curs estiguin definides.
- Que la data passada com a paràmetre estigui entre les dates inicial i final del curs.
- Que la data no correspon a dissabte o diumenge.
- Que la data no es troba en els períodes especificats a la taula *festius* per al curs escolar sol·licitat.

En cas que s'incompleixi una de les condicions anteriors, la funció retornarà el valor *FALS* i el

paràmetre RSP el motiu pel qual la data no és vàlida. En cas contrari retornarà *TRUE* i RSP el valor 'OK'. La signatura és

```
DATA_VALIDA(curs_escolar_ IN VARCHAR2, data_ IN DATE, RSP OUT
STRING):BOOLEAN
```

### Professor vàlid

Per a diferents procediments necessitarem saber si un NIF de professor és correcte, i si és correcte per a un curs escolar en concret.

La signatura de la funció és

```
PROFESSOR_VALID(NIF_ VARACHAR2, curs_escolar_ VARCHAR2, RSP OUT
STRING) : BOOLEAN
```

El paràmetre NIF\_ correspon al NIF del professor, *curs\_escolar\_* al curs que es vol comprovar. Si *curs\_escolar\_* és nul, el procediment comprova únicament si el professor és a la taula *professors* de la base de dades. Si el curs no és nul comprova si el professor a més de ser a la base de dades està assignat a alguna assignatura per al curs en qüestió, consultant la taula *submateries*.

### Amonestació vàlida

Aquesta funció comprova si un número d'amonestació és vàlid per a un curs escolar concret. Aquesta funció no comprova si el curs escolar és vàlid, ja que aquesta comprovació ja s'ha fet abans en els procediments que utilitzen aquesta funció.

La signatura de la funció és

```
AMONESTACIO_VALIDA(amon_ INTEGER, curs_escolar_ VARCHAR2, RSP OUT
STRING):BOOLEAN
```

La funció torna *TRUE* si a la taula *amonestacions* existeix una amonestació amb el número *id* igual a *amon\_* i corresponent al curs escolar *curs\_escolar\_*, i retorna *FALSE* en cas contrari. Serveix per comprovar que es pot inserir un valor a la taula *amonestacions\_alumne* que faci referència a aquesta amonestació.

## 5.4 Procediments

Als procediments on un paràmetre es correspon directament amb el camp d'una taula que s'ha d'utilitzar s'ha agafat com a norma utilitzar el mateix nom del camp acabat en guió baix. Així, per exemple, nom d'un paràmetre que es correspon al camp *exp* de la taula *alumnes* serà *exp\_*.

Als diferents procediments s'han afegit unes variables comuns per a mantenir actualitzada de forma sistemàtica la taula LOG\_PROC. Les variables són LOGPRO, PARAM\_IN i PARAM\_OUT, totes del tipus VARCHAR2. LOGPRO emmagatzema el nom del procediment, PARAM\_IN una cadena amb el nom dels paràmetres d'entrada i el seu valor convertit a cadena alfanumèrica si és necessari, i PARAM\_OUT la part corresponent amb els paràmetres de sortida. A tots els procediments s'inclou la següent instrucció pel cas de que l'execució es dugui a terme sense errors:

```
INSERT INTO LOG_PROC VALUES (SYSDATE,LOGPRO,PARAM_IN,PARAM_OUT);
```

Alguns dels procediments consisteixen bàsicament en fer un INSERT a una taula, l'avantatge

que comporta fer ús d'aquests procediments dins un programa és el control d'errors que suposa comprovar que el paràmetre RSP de sortida torni amb el valor OK si el procediment s'ha executat correctament. En cas que no sigui així, en molts casos el missatge de resposta ajuda a detectar els errors que s'han donat. En alguns casos són missatges d'usuari generats pel procediment, en altres RSP recull el missatge d'error generat per l'Oracle.

### Alta curs escolar

La seva funció és donar d'alta cursos escolars. Es comprova que el curs no existeixi i que la data inicial sigui anterior a la final. La seva signatura és

ALTA\_CURS\_ESCOLAR

(curs\_escolar\_ VARCHAR2, data\_inicial\_ DATE, data\_final\_ DATE, RSP OUT STRING)

Possibles missatges d'error:

RSP = ERROR la data inicial ha de ser anterior a la data final

RSP = ERROR ja existeix el curs escolar

### Baixa curs escolar

La seva funció és donar de baixa cursos escolars. Es comprova que el curs existeixi. La seva signatura és

BAIXA\_CURS\_ESCOLAR

(curs\_escolar\_ VARCHAR2, RSP OUT STRING)

Possibles missatges d'error:

RSP = ERROR no existeix el curs escolar

### Alta grup

Procediment per donar d'alta grups dins un curs escolar. La signatura és

ALTA\_GRUP

(curs\_escolar\_ VARCHAR2, nivell\_ VARCHAR2, grup\_ VARCHAR2, RSP OUT STRING)

Abans de la inserció a la base de dades es comprova que el curs escolar estigui creat, si no és així es dona el missatge RSP=ERROR no existeix el curs escolar .

### Baixa grup

Dona de baixa un grup d'un curs escolar. Abans d'executar l'ordre DELETE es comprova que el grup realment existia. La seva signatura és

BAIXA\_GRUP

(curs\_escolar\_ VARCHAR2, nivell\_ VARCHAR2, grup\_ VARCHAR2, RSP OUT STRING)



## Alta professor

Dona d'alta professors. No queden assignats a cap curs escolar ni grup, això es fa amb les submatèries. Es comprova que no hi hagi cap entrada amb el mateix NIF. La signatura és

```
ALTA_PROFESSOR(nif_ , nom_ , llinatge1_ E, llinatge2_ ,
departament_ , adreça_ , poblacio_ , telefon1_ , telefon2_ , RSP OUT string)
```

## Baixa professor

Dóna de baixa un professor. Prèviament es comprova que el NIF és a la taula *professors*. La signatura és

```
PROCEDURE BAIXA_PROFESSOR (nif_ VARCHAR2, RSP OUT STRING)
```

## Alta alumne nou

Per donar d'alta un alumne se suposa que no existeix al sistema, i se l'ha de matricular. Per tant se li assigna un número d'expedient nou, que serà consecutiu a l'expedient més alt existent. Al mateix procediment se li especifica a quin curs escolar, nivell i grup se'l matricula. El grup pot ser nul, ja que es podria matricular pendent d'assignar un grup, però el curs i nivell escolar no. Tampoc poden ser nuls el nom i primer llinatge de l'alumne.

Aquest procediment insereix un registre a la taula *alumnes*, i un altre a la taula *alumnes\_matricula*. A la taula *alumnes* només s'insereix la informació referent a nombre d'expedient, nom i llinatges. Per introduir més dades s'ha de combinar amb el mètode modifica alumne.

El mètode retorna un error error si el curs escolar no existeix. La signatura és

```
ALTA_ALUMNE_NOU
(nom_ , llinatge1_ , llinatge2_ , curs_escolar_ , nivell_ , grup_ ,
n_exp OUT INTEGER, RSP OUT )
```

## Alta alumne existent

En cas que es vulgui matricular un alumne que ja existeix al sistema no s'han d'introduir les dades de nou. Aquest cas es donarà per alumnes que han estat matriculats en cursos anteriors, o alumnes dels que s'ha fet una baixa per efectuar un canvi de matrícula.

Per aquest procediment s'especifica quin és l'alumne donant el seu número d'expedient, i a quin curs escolar, nivell i grup se'l vol matricular.

## Modifica dades alumne

Aquest procediment permet modificar tots els camps excepte el número d'expedient, que és el camp clau, de la taula *alumnes*. Els paràmetres són tots els camps que apareixen a la taula alumne a més d'un camp numèric anomenat *opcio* que agafa els valors 1 o 2. Aquest camp permet especificar què fer amb els valors *null*. El número d'expedient és l'únic paràmetre obligatori.

L'opció per defecte del paràmetre *opcio* és 1. Si es deixa 1 o buit el procediment únicament modificarà els camps corresponents als paràmetres que no siguin nuls en el moment de fer la crida

per al registre amb número d'expedient especificat.

Si el valor de *opcio* és 2, els paràmetres no nuls modificaran els valors del registre i els paràmetres nuls esborraran els camps corresponents.

## Baixa Alumne

Els alumnes no s'esborren de la base de dades. Si volem donar de baixa un alumne ho farem d'un curs escolar en concret, i serà l'entrada de la taula *alumnes\_matricula* la que s'eliminarà.

## Nova amonestació

El procediment *nova\_amonestacio* serveix per crear una nova entrada a la taula *amonestacions*. Abans de fer la inserció es fan diferents comprovacions:

- El curs escolar és vàlid
- La data és vàlida per al curs escolar
- El professor imparteix classes al curs

La signatura del procediment és

NOVA\_AMONESTACIO((curs\_escolar\_ VARCHAR2, tipus\_ VARCHAR2, professor VARCHAR2, data\_ DATE, hora\_ INTEGER, id\_ OUT INTEGER, RSP OUT STRING )

El paràmetre de sortida *id\_* retorna el valor del camp *id* per a la nova amonestació creada, i permet fer posteriors assignacions a alumnes de l'amonestació creada.

## Modifica amonestacio

Aquest procediment permet modificar els camps no clau de la taula *amonestacions*. La signatura és

MODIFICA\_AMONESTACIO(id\_ INTEGER, curs\_escolar\_ VARCHAR2, tipus\_ VARCHAR2, professor VARCHAR2, data\_ DATE, hora\_ INTEGER, obs\_ VARCHAR2, opcio INTEGER, RSP OUT NOCOPY STRING )

El paràmetre *opcio* permet establir l'opció d'ignorar els valors null dels paràmetres (*opcio*=1 o null), o esborrar els camps deixats com a null (*opcio*=2), tal com s'ha descrit al procediment *modificar\_dades\_alumne*. Aquesta opció afecta només als camps *tipus\_* i *observacions*, ja que els altres camps no poden ser buits pel disseny de la taula *amonestacions*.

## Amonest afegir alumne

Una vegada creada una amonestació a la taula *amonestacions*, es pot relacionar amb un o més alumnes inserint l'entrada corresponent a la taula *amonestacions\_alumne*. La signatura és

AMONEST\_AFEGIR\_ALUMNE(curs\_escolar\_ VARCHAR2, exp\_ INTEGER, amo\_id\_ INTEGER, data\_com DATE, tret\_ CHAR, obs\_ VARCHAR2, RSP OUT STRING)

Els paràmetres *curs\_escolar\_* i *amo\_id\_* serveixen per identificar l'amonestació, *exp\_*

identifica l'alumne que es vol afegir a l'amonestació. Es fan les comprovacions pertinents per assegurar que l'amonestació existeix i que l'alumne està matriculat al curs escolar. També es comprova que la data de comunicació no és anterior a la data de l'amonestació i que correspon a un dia lectiu. Donam per suposat que les comunicacions als pares es fan des del lloc de feina.

### Nova amonestació alumne

Aquest procediment permet en una sola instrucció crear una amonestació i assignar-la a un alumne. Per això internament fa una crida primer al procediment *nova\_amonestacio* i a continuació al procediment *amonest\_afegir\_alumne*. Els paràmetres han de permetre fer les crides als dos procediments:

```
NOVA_AMONESTACIO_ALUMNE(curs_escolar_ VARCHAR2, tipus_ VARCHAR2,
professor_ VARCHAR2, alum_ INTEGER, data_ DATE, hora_ INTEGER, obs_amo_
VARCHAR2, data_com_ DATE, tret_ CHAR, obs_alum_ VARCHAR2, id_ OUT INTEGER, RSP
OUT STRING )
```

Si hi ha errors al primer procediment *nova\_amonestacio* el segon *amonest\_afegir\_alumne* ja no s'executa. En cas que hi hagi errors al segon procediment, el primer s'haurà executat correctament. Per evitar altres errors, en aquest cas el procediment esborrarà l'amonestació creada abans d'acabar la seva execució.

Aquest procediment no crea una entrada pròpia a la taula LOG\_PROC, ja que els dos procediments cridats ja queden enregistrats amb tots els paràmetres necessaris.

### Elimina amonestació alumne

Aquest procediment elimina l'entrada corresponent de la taula *amonestacions\_alumne*, no elimina l'entrada de la taula *amonestacions* que té la informació posada pel professor. Elimina la relació que hi ha establerta entre una amonestació i un alumne matriculat al curs escolar.

```
ELIMINA_AMONESTACIO_ALUMNE
```

```
(curs_escolar_ VARCHAR2, exp_ INTEGER, amo_id_ INTEGER, RSP OUT STRING)
```

### Nova sanció

El procediment *nova\_sancio* permet introduir noves sancions a la taula *sancions*. En cas de que es tracti d'una sanció amb una càrrega econòmica, el camp *pendent* es posarà automàticament amb el mateix valor que la quantia de la sanció. En cas que el paràmetre *tipus\_* es deixi buit, es considerarà que es tracta d'una sanció personalitzada i al camp *tipus\_sancio* es posarà amb el valor 'pers'.

Cada vegada que s'insereixi una sanció s'haurà d'actualitzar el camp *comp* de la taula *amonestacions\_alumne*, per marcar les amonestacions que s'han tingut en compte per generar la sanció.

Això es pot fer inserint les sancions sempre mitjançant un procediment, o amb un disparador que actualitzi el camp cada vegada que s'actualitzi la taula *sancions*.

## Elimina sanció

Aquest procediment elimina l'entrada corresponent a la taula *sancions*. Es comprova que el curs és vàlid i que la sanció existeix. La signatura és

ELIMINA\_SANCIO

(curs\_escolar\_ VARCHAR2, id\_sancio\_ INTEGER, RSP OUT STRING)

## Sanció automàtica

Cada vegada que es posi una nova amonestació a un alumne, la base de dades ha de comprovar si s'han acumulat prou amonestacions com per generar una nova sanció. Per poder fer aquesta comprovació s'ha de tenir en compte el camp *comp* de la taula *amonestacions\_alumne*, que indica per a cada amonestació posada a un alumne si ja s'ha tengut en compte per a generar una sanció. Dit d'una altra manera, es comptabilitzen les amonestacions que no han estat ja sancionades, per evitar que a un alumne se'l castigui dues vegades pels mateixos fets.

Les sancions generades d'aquesta manera han de quedar pendents per revisar pels caps d'estudis, que finalment aplicaran la sanció estudiant cada cas de forma individual.

El procediment *sancions\_automatiques* es crida directament des del procediment *amonest\_afegir\_alumne*, de forma que després d'afegir una amonestació es comprova si és del tipus que apareix a alguna de les regles de sanció automàtica i després si el nombre de amonestacions d'aquest tipus sense computar és suficient per a generar una nova sanció. Si és així aquesta és generada indicant que és resultat de sanció automàtica, i les amonestacions que han servit per generar-la es marquen com a computades.

## 5.5 Procediments mòdul estadístic

Per al mòdul estadístic la classificació de procediments s'ha fet separada per ser una part bastant específica del projecte. Aquests procediments serveixen per inicialitzar i per reparar totes les entrades a taules del mòdul estadístic. No estan dissenyats per mantenir les taules actualitzades en tot moment, ja que això es farà amb disparadors que afectaran únicament a les dades que s'hagin de modificar. Si les dades s'introdueixen al sistema de forma correcta, amb els disparadors compilats prèviament, no haurà necessitat d'executar aquests procediments en cap moment.

Els diferents procediments són els següents:

### Inicia estadístiques

Amb aquest procediment es creen totes les entrades de la taula *estadistiques*, amb tots els seus valors globals per a un curs escolar concret, com s'ha descrit a l'apartat de descripció de les taules de la base de dades.

A més es va cridant al procediment *inicia\_estadistiques\_nivell* per a cada un dels nivells que s'hagin creat per aquell curs escolar, i als procediments *calcula\_alumne\_mes\_amonestat*, *calcula\_alumne\_mes\_sancionat* i *calcula\_prof\_mes\_amonestador*. D'aquesta manera, cridant el procediment *inicia\_estadistiques* les cinc taules que guarden la informació del mòdul estadístic queden actualitzades.

També es creen les entrades de la taula *estadist\_alumnes*, on es guarden el nivell i nombre de sancions i amonestacions de cada alumne. És l'únic procediment del mòdul estadístic que enregistra

una entrada a la taula *log\_proc*, ja que la resta de procediments es criden des d'aquest.

En un funcionament normal de l'aplicació no serà necessari executar aquest procediment, ja que les taules estadístiques es van mantenint sempre actualitzades mitjançant disparadors.

### Inicia estadístiques nivell

Amb aquest procediment es crea una fila corresponent al curs escolar i nivell que s'han passat com a paràmetre a la taula *estadistiques\_nivell*. Es calculen el valor de nombre de professors que imparteixen el nivell, nombre d'alumnes, d'amonestacions, sancions, nombre d'alumnes amb alguna amonestació, amb alguna sanció, i nombre de professors que han posat alguna amonestació.

També es creen entrades a *estadist\_professors*, però aquí es creen entrades per a tots els professors i no només els que hi fan classe. Això és perquè es contempla la possibilitat de què un professor pugui posar una amonestació a un alumne encara que no sigui professor seu.

### Calcula alumne més amonestat

Aquest procediment consulta la taula *amonestacions\_alumne* per esbrinar quins són els alumnes de cada nivell que tenen el nombre més alt d'amonestacions, i també quin són els que tenen el nombre més alt a nivell de tot el centre.

Els alumnes amb nombre més alt a nivell de centre tendran dues entrades a la taula, una a nivell de centre i una altra per al seu nivell educatiu.

### Calcula alumne més sancionat

Aquest procediment funciona de forma anàloga a l'anterior, però comptabilitza els alumnes amb el major nombre de sancions per nivell.

### Calcula professor més amonestador

Per a cada nivell i per al centre es comptabilitzen els professors que han posat un major nombre d'amonestacions. Com als dos procediments anteriors, en cas d'empat hi haurà més d'una entrada per nivell o pel centre.

## 5.6 Disparadors

El mòdul estadístic s'ha de mantenir sempre actualitzat. Per això els disparadors són ben adequats. El primer que s'ha de fer és veure quins són els canvis a les taules de la BD que haurien d'afectar al mòdul estadístic, i veure quins canvis s'han de fer a cada cas. No és necessari calcular cada vegada tots els valors que surten a les taules estadístiques. Executar *INICIA\_ESTADISTIQUE*S a cada canvi seria possible, però molt poc eficient en el moment que el volum de dades fos gran.

Els esdeveniments que han de provocar canvis en alguna taula del mòdul estadístic són:

- Creació o supressió d'un curs escolar
- Creació d'un grup nou que suposi un nou nivell educatiu per al curs en què es crea
- Alta/baixa d'alumnes
- Alta de professors a un nivell educatiu.

-Inserció de noves amonestacions a alumnes

-Inserció de noves sancions.

Els disparadors s'activen amb insercions o supressions a les taules, és indiferent que aquestes es duguin a terme directament amb instruccions SQL o mitjançant els procediments presentats prèviament.

### Afegir curs

Després de crear un curs nou s'ha d'afegir l'entrada corresponent a la taula *estadistiques*. Hi havia la possibilitat de crear una entrada per nivell a la taula *estadistiques\_nivell*, però finalment s'ha optat per no fer-ho i que es creen només les entrades corresponents als nivells que tenen algun grup donat d'alta. No sempre s'imparteixen tots els nivells a tots els cursos escolars.

### Elimina curs

En cas de suprimir un curs, s'eliminaren les files corresponents a les taules estadistiques i estadistiques\_nivell. A la resta de taules no hi hauria d'haver informació relativa a aquest curs, ja que per esborrar un nivell no hi ha d'haver registres d'altres taules que l'apunten per claus foranies.

### Afegir grup

Després de la creació d'un nou grup es comprovarà si existeix l'entrada corresponent al seu nivell a la taula *estadistiques\_nivell*. En cas que no existeixi aquesta serà creada. Si a un centre no s'imparteixen tots els nivells educatius s'evita tenir files creades que no són necessàries.

### Elimina grup

En cas de supressió d'un grup, un disparador comprovarà si hi ha més grups per al mateix curs escolar i nivell. En cas de que no hi hagués més grups del nivell aquest s'eliminarà de la taula *estadistiques\_nivell*.

### Afegir submatèria

Les submatèries creades s'han d'assignar a un professor. D'aquesta manera es pot considerar que un professor està donat d'alta a un nivell per a un curs escolar si hi ha alguna submatèria que els relaciona.

Quan es crea una submatèria el disparador comprova si el professor està donat d'alta per al curs i nivell, a la taula *estadist\_professors*, i si no és així crea les entrades corresponents. A més s'actualitzaran els valors de *n\_professors* a *estadistiques* i *estadistiques\_nivell*, si s'escau.

### Afegir alumne

En cas de crear-se una nova matrícula per a un curs escolar, amb una inserció a la taula *alumnes\_matricula*, s'hauran de fer diferents operacions com ara actualitzar el nombre d'alumnes del centre (*n\_alumnes* de la taula *estadistiques*), el nombre d'alumnes del nivell (*n\_alumnes* de la taula *estadistiques\_nivell*), i crear l'entrada corresponent a l'alumne per al curs escolar a la taula *estadist\_alumnes*.

El fet de crear una nova entrada a la taula alumnes no té cap efecte sobre les taules

estadístiques.

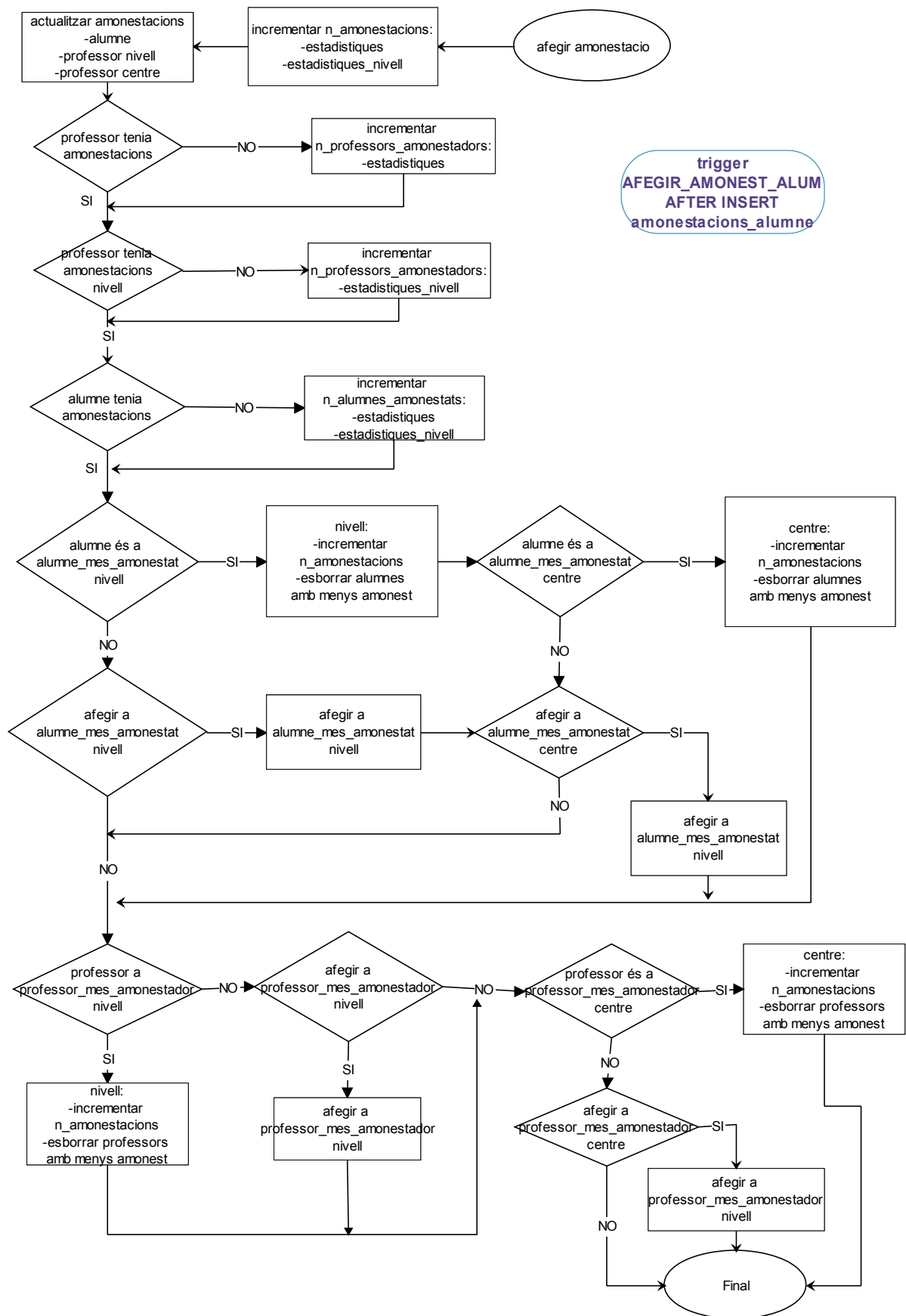
### Elimina alumne

En cas de eliminar les dades d'una matrícula les taules estadístiques han de quedar en el mateix estat que abans de la inserció. S'ha de decrementar el nombre d'alumnes al curs i al nivell, i s'ha d'eliminar l'entrada a *estadist\_alumnes* corresponent a l'alumne.

### Afegir amonestació

Només s'activa en cas d'inserció en la taula *amonestacions\_alumne*. El fet d'afegir una amonestació a un alumne afecta a diferents camps de les taules estadístiques. S'han de modificar el nombre d'amonestacions per al centre i per al nivell educatiu; s'ha de comprovar si hi ha un augment al nombre d'alumnes amonestats i professors amonestadors, tant al curs com a tot el centre. A més s'ha de comprovar si hi ha canvis a les taules amb els alumnes més amonestats i professors més amonestadors, una altra vegada també a nivell de curs com de tot el centre.

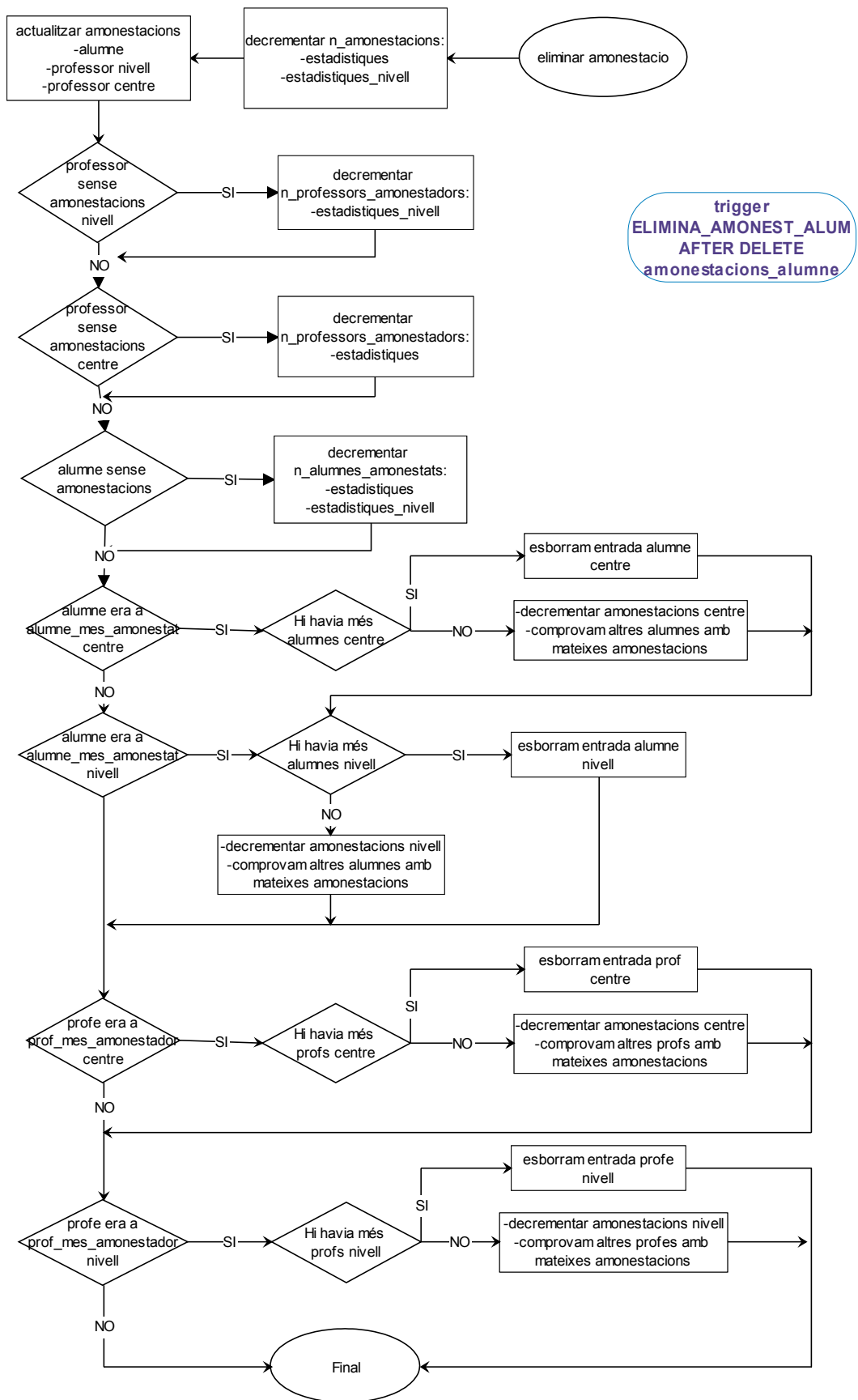
Per la seva complexitat, s'ha trobat necessari elaborar un diagrama de flux per a l'elaboració del script corresponent a aquest disparador, que se mostra a continuació:





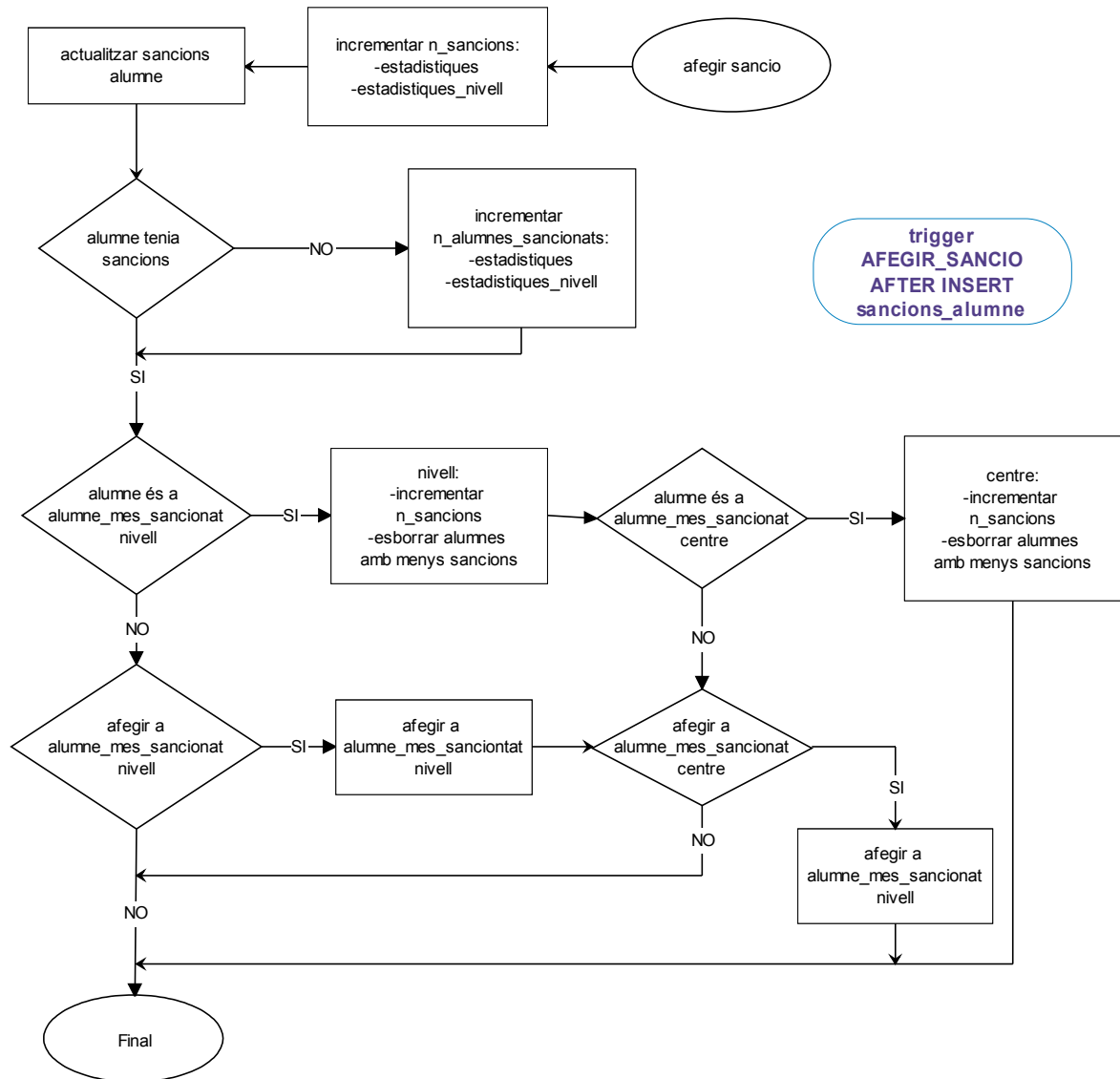
## **Elimina amonestació alumne**

El següent diagrama mostra les operacions fetes a les taules estadístiques després d'eliminar una amonestació:



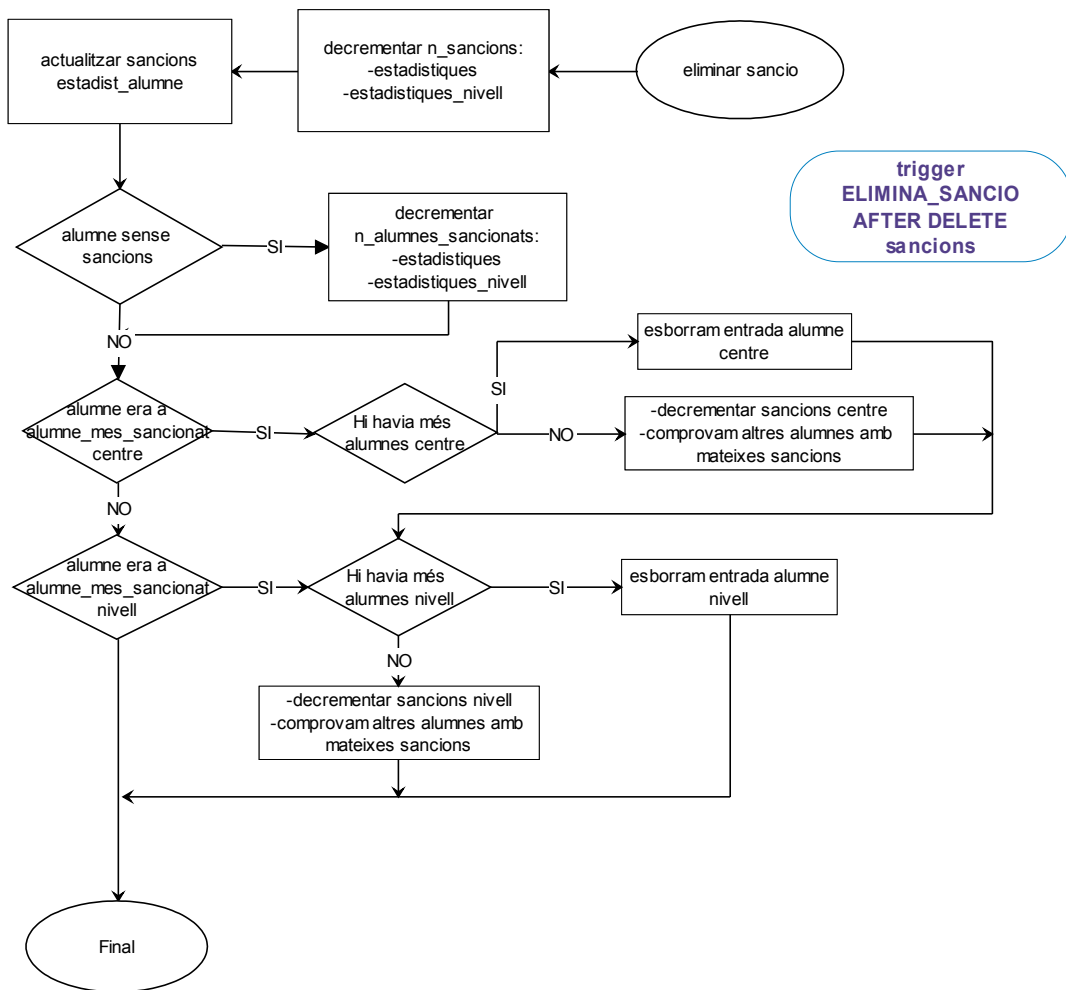
### Afegir sanció

Després d'afegir una sanció s'han d'actualitzar els camps *n\_sancions* i *n\_alumnes\_sancionats*, tant a la taula *estadistiques* com a *estadistiques\_nivell*. També s'ha de comprovar si hi ha d'haver actualitzacions a la taula *alumne\_mes\_sancionat*. El funcionament del trigger serà semblant al executat després d'inserir amonestació, però sense la complicació addicional d'haver de mantenir també informacions referents a professors. El diagrama de flux queda



### Elimina sanció

Finalment, les taules també s'han de modificar després d'eliminar una sanció:



## 6 Jocs de proves

El joc de proves es presenta al document *provesPROCEDIMENTS.sql*, i consisteix en un bloc executable amb totes les proves a dins. També es poden anar provant els diferents procediments executant-los directament des del SQL-Developer, i posant com a paràmetres els que apareixen al document.

No s'han inclòs proves per al mòdul estadístic, ja que aquest es va actualitzant de manera independent sense crides a cap procediment. En tot cas es poden esborrar manualment les taules del mòdul estadístic per a continuació executar *INICIA\_ESTADISTIQUES(curs\_escolar\_, RSP)*.

## 7 Presentació virtual

Un dels productes finals del TFC és la presentació virtual. Aquesta no afegeix informació addicional a la que ja surt en aquesta memòria, sinó que està plantejada com un document de síntesi on es fa un resum dels aspectes més rellevants del treball realitzat. Està centrat principalment en el funcionament dels diferents procediments i funcions de la base de dades, i a explicar com s'ha

plantejat el mòdul estadístic.

És una manera de mostrar les primeres passes que es poden donar amb aquesta aplicació, de forma que un usuari podria començar a utilitzar-la seguint les seves explicacions.

En concret s'intenta explicar la forma de guardar les dades corresponents a calendari i horaris, a explicar el concepte de submatèria i la seva utilitat, i especialment a explicar el disseny del mòdul estadístic.

La presentació, igual que la memòria i la planificació inicial s'ha elaborat utilitzant el paquet ofimàtic OpenOffice de Oracle de lliure distribució, i es distribueix en format pdf.

## 8 Glossari.

**Amonestació:** Avís per escrit fet a un o més alumnes i als seus pares com a conseqüència del seu mal comportament. Pot comportar una sanció o no. La reiteració de tres amonestacions comporta una sanció en tots els casos. Les amonestacions són imposades per professors del centre.

**Curs escolar:** Cada un dels períodes en què està organitzat l'ensenyament reglat, on els alumnes es matriculen per a un determinat nivell i tenen un horari i equip de professors determinats. A cada curs escolar s'ha de fer una nova matrícula, assignació de grups i horaris, assignació de professors i membres de l'equip directiu i altres tasques. Cada curs escolar ha de constar de 176 dies lectius entre setembre i juny, tenint en compte els dies festius i períodes de vacances.

**Festius:** Tenen consideració de festius tots els dies entre dilluns i divendres que queden entre les dates inicial i final d'un curs escolar, però en els quals no es farà classe. Es podran especificar dies festius a nivell estatal o de la comunitat, festes locals o dies de lliure designació del centre, així com períodes de vacances com ara nadal o pasqua.

**Matèria:** Cada una de les assignatures que poden cursar els alumnes en un curs escolar, venen determinades pel currículum oficial de la Conselleria d'Educació.

**Nivell:** Fa referència als diferents nivells educatius que formen l'escolarització d'un alumne. L'escolarització obligatòria inclou els nivells de 1r a 4t d'ESO per als instituts de secundària. L'ensenyament postobligatori als instituts pot estar format per 1r i 2n de batxillerat, i per diferents cicles de formació professional de grau mitjà o superior. Per matricular-se a un nivell els alumnes han d'haver cursat els nivells anteriors.

**Pagament:** cada un dels reintegraments de diners que es fan a l'institut per satisfer una sanció que comportava un cost econòmic. La suma dels diferents pagaments ha d'arribar a ser igual a la quantia imposada.

**Sanció:** Càstig imposat a un alumne per part d'un membre de l'equip directiu. Normalment serà conseqüència d'haver rebut una o més amonestacions, segons la seva gravetat. Pot anar acompanyada d'una càrrega econòmica, en cas d'haver-hi danys materials.

**Submatèria:** Cada una de les agrupacions d'alumnes que es fan durant un curs escolar per a impartir una determinada matèria. Cada submatèria anirà associada a un professor i un horari concret.

## 9 Bibliografia.

Per a l'elaboració del treball s'han anat consultant diverses fonts. Aquestes estan agrupades segons la part del treball per la qual han estat consultades:

### **Pautes per a l'elaboració de la memòria:**

-Material web de l'assignatura.

### **Elaboració de scripts:**

-Material de cursos anteriors de la UOC (BDI i BDII)

-Oracle PL/SQL Programming, 2nd edition. Steven Feuerstein, Bill Pribyl Editorial. O'Reilly

## 10 Annexos.

### 10.1 Presentació virtual

La presentació en diapositives s'annexa a l'arxiu *presentacio Lluís Aristondo Arvidsson.odt*

### 10.2 Scripts de la base de dades

Es presenten els scripts als següents arxius:

1. disciplinaDDL.sql per a la creació de taules i vistes.
2. disciplinaTRIGGER.sql per a la creació de disparadors.
3. disciplinaFUNC.sql per a la creació de funcions.
4. disciplinaPROC.sql per a la creació de procediments. Inclou procediments per a funcionament de la BD i per a la reparació de taules del mòdul estadístic.
5. disciplinaDML.sql per a la introducció de registres permanents (necessaris per al funcionament de la BD).
6. DADES\_EXEMPLE.sql per a la introducció de registres d'exemple.
7. ProvesPROCEDIMENTS.sql amb dades per a provar els diferents procediments.



# 10.3 Disseny de la BD

