UNIVERSITAT OBERTA DE CATALUNYA

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

Roberto Jiménez Llahí
ETIG

Ismael Pérez Laguna

12/06/2011

Resum

El treball que es presenta a continuació pretén esser el recull de gran part dels coneixements adquirits en el procés d'aprenentatge en Enginyeria Tècnica en Informàtica de Gestió, així com l'aplicació pràctica dels mateixos. El que es presenta dons es un projecte informàtic, en aquest cas de base de dades, passant per totes les seves etapes.

Començarem amb un pla de treball detallat del projecte, amb totes les fases a desenvolupar ajustat al període de temps que es disposa per completar-ho. Un cop definit el pla de treball i el temps destinat a cadascuna de les tasques, i tenint-les clares, es començarà a treballar sobre els requeriments per construir el disseny, que servirà de base per desenvolupar el els scripts del producte.

Un cop acabat el disseny comença l'etapa d'implementació del producte, en primera instància, amb l'estructura o esquelet de la base de dades, que servirà de base i contenidor de dades dels procediments que interactuaran entre en usuari i les dades que s'aniran omplint. S'estructurà el codi aplicant els coneixements adquirits sobre desenvolupament durant aquests anys d'estudi.

Un cop finalitzada la implementació es realitzarà un test unitari bàsic per comprovar el codi.

Dedicatòria i agraïments

Aquest treball està dedicat a la meva parella, i a tota la meva família, per haver-me engrescat sempre en tots els projectes personals i professionals que he decidit emprendre. La vostra confiança en mi ha estat la font que ha alimentat les meves esperances, dia rera dia, per a no perdre mai l' ànim ni les ganes d' aprendre i per a continuar endavant.

El meu agraïment a tots els que d'alguna manera heu contribuït a aquest èxit personal. Vosaltres sabeu qui sou i fins a on arriba el meu deute.

Gracies també a la part docent de la UOC, i ànims a tots els estudiants, la constància i l'esforç acaba tenint la seva recompensa.

Índex

Resum III

Dedicatòria i agraïments IV

Índex V

1. Introducció 1

1.1. Descripció del TFC 1

1.2. Objectius generals i específics 2

1.3. Enfocament i mètode seguit 3

1.4. Rols i participants 3

1.5. Planificació del projecte 4

1.5.1. Descomposició estructural de les activitats 5

1.5.2. Planificació amb fites i temporització 6

1.5.3. Diagrama de Gantt 7

1.5.4. Anàlisis de riscos 8

1.6. Valoració econòmica 8

1.7. Documents a entregar 10

1.7.1. PAC1 10

1.7.2. PAC2 11

1.7.3. PAC3 11

1.7.4. Producte final 12

2. Disseny 13

2.1. Disseny de la BBDD 13

2.2. Disseny de la gestió dels Logs 16

2.3. Disseny dels procediments d'ABM 17

2.4. Disseny del mòdul estadístic 19

2.5. Disseny de la gestió d'amonestacions i sancions 21

2.6. Disseny dels procediments de consulta 22

3. Implementació 23

3.1. Creació de les taules 23

3.2. Creació dels procediments d'ABM 24

3.3. Creació dels procediments per a la gestió estadística 25

3.4. Creació dels procediments per a la gestió d'amonestacions i sancions 26

3.5. Creació dels procediments de consultes 28

4. Testing 29

4.1. Carrega inicial de dades 29

4.2. Creació del joc de proves de dades 29

4.2.1. Operacions prèvies 29

4.2.2. Proves dels procediments de Alta, Baixa i Modificació 30

4.2.3. Proves de Amonestacions i Sancions 32

4.2.4. Proves de Estadístiques 33

4.2.5. Comprovació de les consultes 33

4.2.6. Comprovació dels Logs 35

5. Conclusions 36

6. Glossari 37

7. Bibliografia 39

Annexos: Codi de l'aplicació 40

Creació de Taules 40

Creació de procediments de Estadístiques 48

Creació de procediments de Consultes 52

Creació de procediments de Amonestacions i Sancions 54

Creació de procediments de Alta, Baixa i Modificació 65

1. Introducció

1.1. Descripció del TFC

Aquest projecte consisteix en el desenvolupament d'un sistema de gestió de les amonestacions i sancions trobades als centres educatius, així com l'explotació de les dades.

Es pretén guardar la informació dels alumnes matriculats, dels professors, i dels cursos on estan matriculats, així com la informació sobre quins són els tutors de cada grup i els horaris existeixen. Tota aquesta informació es gestionarà a través de cadascun dels diferents instituts d'ensenyament de Catalunya.

El sistema a dissenyar ha de permetre emmagatzemar tota la informació comentada anteriorment i permetre generar les consultes més habituals que es realitzen. A més, la base de dades s'haurà d'encarregar de precalcular i emmagatzemar diversa informació estadística, per a fer accessible determinada informació de caràcter general sense carregar la base de dades.

En primer lloc, es tractaria de realitzar un estudi i anàlisis dels requeriments, per a seguidament dissenyar l'esquelet del sistema. Per realitzar aquest objectiu es seguiran els següent passos:

- Dissenyar la base de dades (Diagrama E/R), fer l'script de creació de taules, índex, etc. necessaris e implementar els procediments emmagatzemats requerits per al manteniment de dades

- Creació de un mecanisme de inicialització de dades de mostra per a la base de dades per tal de poder realitzar proves i simular el funcionament normal del sistema.

A partir d'aquí, es començarà a implementar les funcionalitats que pugui oferir la solució, dotant de disparadors, procediments emmagatzemats i procediments per a consultes, tant bàsiques com estadístiques, al sistema.

1.2. Objectius generals i específics

L'objectiu general del TFC es la realització d'un projecte informàtic que permeti posar en pràctica els coneixements adquirits durant la carrera.

Aquest TFC en concret, es centra sobre tot en els coneixements adquirits en les assignatures de Bases de Dades I y Bases de Dades II.

A més, el fet de que el TFC es realitzi en el SGBD Oracle permetrà posar en pràctica els coneixements en un sistema de gestió de gran presencia en el mercat, àmpliament utilitzat per moltes de les grans companyies dels diversos sectors.

Al llarg del cicle de vida del Projecte, es realitzaran diferents entregues, que seran les següents:

- **PAC1:** Pla de Treball. Te com objectiu arribar a establir una planificació realista.
- **PAC2:** Entrega del model E/R i del la BBDD.
- **PAC3**: En esta PAC s'entrega tot el corresponent a la fase d'implementació, així com la de probes internes.

A més d'aquestes tres entregues d'avaluació continuada, es realitza la següent entrega:

- Entrega final: La entrega final constarà de las següents parts:
    - Memòria.
    - Presentació.
    - Treball pràctic.

1.3. Enfocament i mètode seguit

L'objectiu del pla de treball és definir i detallar en major mesura les tasques, accions i riscos que s'han de portar a terme en el projecte, així com el pes temporal i l'esforç que s'haurà de dedicar a cadascuna d'aquestes tasques per tal de acomplir amb la temporització del projecte.

Degut a la naturalesa del TFC la metodologia seguida durant el cicle de vida del projecte es la que se coneix com "Cicle de Desenvolupament en Cascada".

Un exemple de desenvolupament en cascada es:

1. Anàlisis de requisits
2. Disseny del Sistema
3. Disseny del Programa
4. Codificació
5. Proves
6. Implantació
7. Manteniment

Seguint aquesta guia, el primer pas per afrontar aquest projecte ha sigut realitzar un anàlisis inicial dels requeriments del enunciat i elaborar un pla de treball que permeti aconseguir els objectius del projecte.

En el desenvolupament del projecte s'utilitzaran els coneixements que les assignatures de la carrera ens han aportat, especialment BBDD I, BBDD II, SGBD e Informàtica aplicada a la gestió.

1.4. Rols i participants

Cada participant té un rol determinat.

- Ismael Pérez Laguna (consultor): Realitzarà el paper de empresa client. Les especificacions sobre els punts del document que no quedin clars seran consultades amb ell. Duran les entregues proporciona el feedback suficient per saber la seva impressió sobre la evolució del projecte.

- Roberto Jiménez (alumne): Realitzarà el paper de proveïdor de software a mida. S'encarregarà de dissenyar, implementar i documentar el software. Es mantindrà en contacte amb el client en tot moment per si es necessari especificar més algun punt en concret de l'aplicació.

1.5. Planificació del projecte

A continuació s'indiquen les dades claus d'aquests projecte:

Data inicial del projecte: 03-03-2011

Data final/entrega del projecte: 12-06-2011

Dies naturals: 102

Setmanes fins la entrega final: 15

Estimació d'esforç setmanal: De 8 a 12 hores

Les entregues es realitzen durant les següents dades:

- PAC1 s'entrega el dia 20/03/2011.
- PAC2 s'entrega el dia 17/04/2011.
- PAC3 s'entrega el dia 15/05/2011.

1.5.1. Descomposició estructural de les activitats

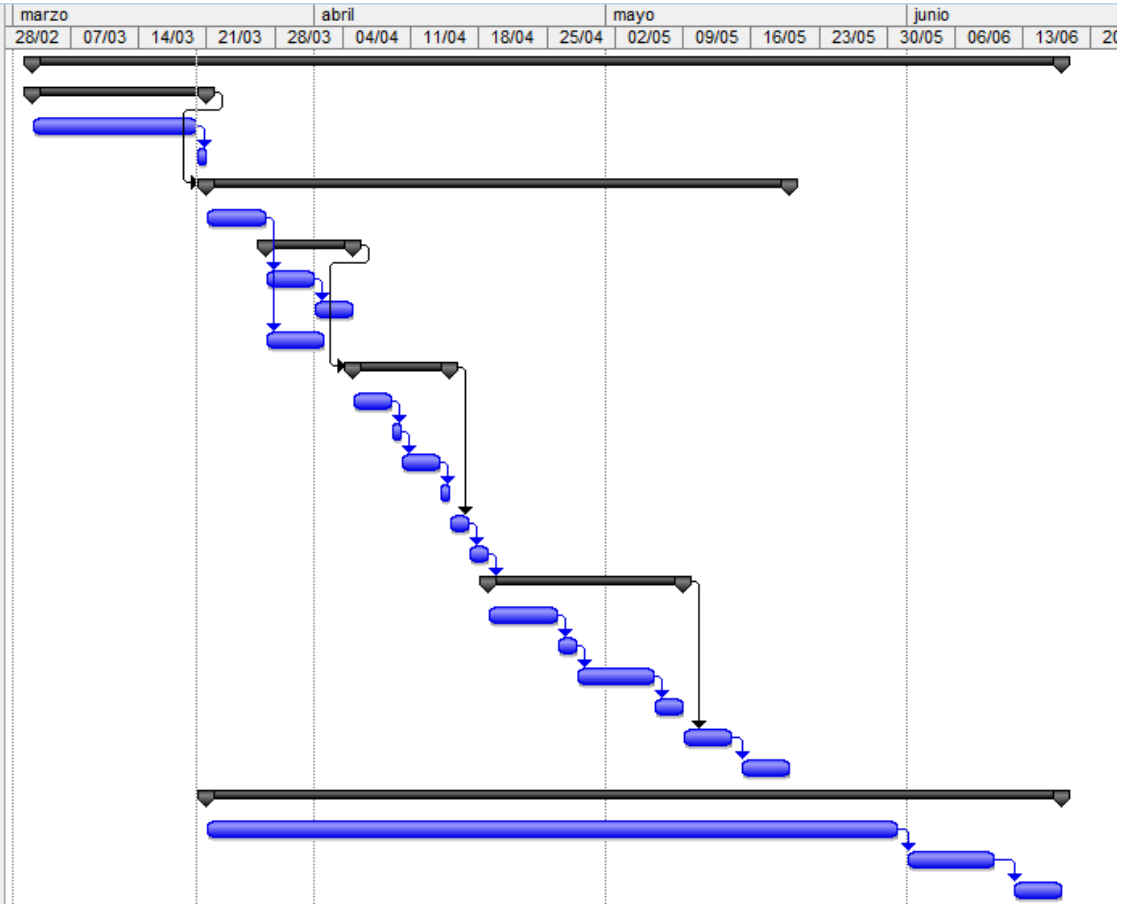
Codi de la activitat	Nom de la activitat de nivell 1	Nom de la activitat de nivell 2	Nom de la activitat de nivell 3
1	Planificació TFC		
1.1		Elaboració Pla de Treball TFC	
1.2		Entrega Pla de Treball TFC (PAC 1)	
2	Desenvolupament BBDD		
2.1		Revisió dels requeriments BBDD	
2.2		Disseny BBDD	
2.2.1			Disseny conceptual BBDD
2.2.2			Disseny lògic BBDD
2.3		Instal·lació i Configuració SGBD ORACLE	
2.4		Construcció esquelet BBDD (PAC2)	
2.4.1			Construcció scripts creació BBDD
2.4.2			Proves unitàries scripts creació BBDD
2.4.3			Construcció procediments emmagatzemats ABM a la BBDD
2.4.4			Proves unitàries procediments emmagatzemats ABM a la BBDD
2.5		Revisió Final Documentació i memòria BBDD (PAC2)	
2.6		Entrega BBDD (PAC 2)	
2.7		Construcció i Proves BBDD (PAC3)	
2.7.1			Construcció disparadors BBDD
2.7.2			Proves unitàries disparadors BBDD Mòdul estadístic
2.7.3			Construcció procediments emmagatzemats per a la explotació de la BBDD
2.7.4			Proves unitàries procediments emmagatzemats per a la explotació de la BBDD
2.8		Revisió Final Documentació i Entregues BBDD	
2.9		Entrega BBDD (PAC3)	
3	Elaboració Memòria i Presentació TFC (ENTREGA FINAL)		
3.1		Elaboració Memòria	
3.2		Elaboració Presentació TFC	
3.3		Entrega Memòria i Presentació TFC (ENTREGA FINAL)	

1.5.2. Planificació amb fites i temporització

Setmana	Dates	Activitat	Esdeveniment
1	03 al 06 de Març	1.1	
2	07 al 13 de Març	1.1	
3	14 al 20 de Març	1.1 i 1.2	Dia 20 entrega PAC1
4	21 al 27 de Març	2.1	
5	28 al 03 de Abril	2.2.1, 2.2.2 i 2.3	
6	04 al 10 de Abril	2.4.1 i 2.4.2	
7	11 al 17 de Abril	2.4.3, 2.4.4, 2.5 i 2.6	Dia 14 entrega PAC2
8	18 al 24 de Abril	2.7.1	
9	25 al 01 de Maig	2.7.2 i 2.7.3	
10	02 al 08 de Maig	2.7.3 i 2.7.4	
11	09 al 15 de Maig	2.8 i 2.9	Dia 15 entrega PAC3
12	16 al 22 de Maig	2.9	
13	23 al 29 de Maig	Memòria i últims retocs	
14	30 al 05 de Juny	Memòria i últims retocs	
15	06 al 12 de Juny	Memòria i últims retocs	Dia 12 Entrega final
16	13 al 19 de Juny	Preparació tribunal	
17	20 al 26 de Juny	Tribunal virtual	Tribunal virtual

1.5.3. Diagrama de Gantt

	Nombre de tarea	Comienzo
1	**TFC**	**jue 03/03/11**
2	**1 Planificació TFC (PAC 1)**	**jue 03/03/11**
3	1.1 Elaboració Pla de Treball TFC	jue 03/03/11
4	1.2 Entrega Pla de Treball TFC (PAC 1)	dom 20/03/11
5	**2 Desenvolupament BBDD (PAC 2 i PAC3)**	**lun 21/03/11**
6	2.1 Revisió dels requeriments BBDD	lun 21/03/11
7	**2.2 Disseny BBDD**	**dom 27/03/11**
8	2.2.1 Disseny conceptual BBDD	dom 27/03/11
9	2.2.2 Disseny lògic BBDD	vie 01/04/11
10	2.3 Instal·lació i Configuració SGBD ORACLE	dom 27/03/11
11	**2.4 Construcció esquelet BBDD (PAC2)**	**mar 05/04/11**
12	2.4.1 Construcció scripts creació BBDD	mar 05/04/11
13	2.4.2 Proves unitàries scripts creació BBDD	sáb 09/04/11
14	2.4.3 Construcció procediments emmagatzemats ABM a la BBDD	dom 10/04/11
15	2.4.4 Proves unitàries procediments emmagatzemats ABM a la BBDD	jue 14/04/11
16	2.5 Revisió Final Documentació i memòria BBDD (PAC2)	vie 15/04/11
17	2.6 Entrega BBDD (PAC 2)	dom 17/04/11
18	**2.7 Construcció i Proves BBDD (PAC3)**	**mar 19/04/11**
19	2.7.1 Construcció disparadors BBDD	mar 19/04/11
20	2.7.2 Proves unitàries disparadors BBDD Mòdul estadístic	mar 26/04/11
21	2.7.3 Construcció procediments emmagatzemats per a la explotació de la BBDD	jue 28/04/11
22	2.7.4 Proves unitàries procediments emmagatzemats per a la explotació de la BBDD	vie 06/05/11
23	2.8 Revisió Final Documentació i Entregues BBDD	lun 09/05/11
24	2.9 Entrega BBDD (PAC3)	dom 15/05/11
25	**3 Elaboració Memòria i Presentació TFC (ENTREGA FINAL)**	**lun 21/03/11**
26	3.1 Elaboració Memòria	lun 21/03/11
27	3.2 Elaboració Presentació TFC	mié 01/06/11
28	3.3 Entrega Memòria i Presentació TFC (ENTREGA FINAL)	dom 12/06/11



1.5.4. Anàlisis de riscos

Al no ser un projecte laboral a temps fixat, sinó que el temps de dedicació va en funció del temps lliure del recurs, els riscos sobre la temporització augmenten notablement.

Es procedeix a llistar el riscos que poden afectar la temporització del projecte:

- Motius laborals: Viatges per motius laborals.
- Motius personals: Esdeveniments familiars. Cap de setmana fora.
- Motius tècnics: Problemes de hardware, o per corrupció de la informació.

Donada la peculiaritat del projecte, en el que al temps de treball fa referència, el pla de contingència depèn del tipus de problemes a enfocar.

Si hi hages un problema per motius tècnics, al disposar de diverses copies de seguretat i diversos ordinadors on continuar amb el desenvolupament, tindríem el pla de contingència definit.

Les copies de seguretat es realitzaran al finalitzar cada jornada de treball, desant la copia en un dispositiu extern respecte al equip de treball, ja bé sigui un llapis USB o un disc dur extern.

Si algun risc acaba afectant el transcurs normal del projecte, problemes laborals, familiars,... la solució i el pla de contingència es ampliar el temps de treball a costa d'hores nocturnes. Això es donat a que no es pot fer un balanç de carrega a altres recursos, donat que només es té recurs disponible.

1.6. Valoració econòmica

Es tracta únicament d'una estimació i el nombre de hores previstes per al desenvolupament del projecte pot variar.

La estimació global d'esforços considera 15 setmanes. Considerant un esforç comprés entre 8 y 12 hores setmanals, considerem que el nombre de hores assignades a aquest projecte pot oscil·lar entre 120 y 180 hores en total.

Segons *Infojobs Trends Salarios*[1], el salari mig d'un analista de bases de dades es trova , en els últims mesos, en torn als 30.000 euros bruts anuals, suposant que es treballa sota el

[1] Pàgina de resultats de la recerca:
http://salarios.infojobs.net/resultados.cfm?sueldo=+bases+de+datos&o_id=2

"Convenio de Empresas de Ingeniería y Oficinas de Estudios Técnicos"[2] que marca una jornada laboral de 1806 hores, es troba que el preu brut per hora mig està en torn als 16'6 euros.

A aquest cost s'hauria de afegir el cost de la llicencia d'Oracle 10g. Utilitzant http://www.dbazone.com/products.html com a pagina de consulta tenim que el cost de una llicencia estàndard (Standard Edition One) ronda els 4000 euros ($5800). Al cost d'aquesta llicencia es té d'afegir el preu per cada un dels usuaris que es vulguin tindre en el sistema, el preu del qual oscil·la sobre els 125 euros ($180).

Concepte	Quantitat	Preu (euros)	Total (euros)
Personal	120 hores	16'6	1992
Llicencia d'Oracle	1 llicencia	4000	4000
Usuaris finals que accediran al sistema	2 usuaris	125	250
		Total	**6.242 euros**

Taula 1 Valoració econòmica optimista

Concepte	Quantitat	Preu (euros)	Total (euros)
Personal	180 hores	16'6	2988
Llicencia d'Oracle	1 llicencia	4000	4000
Usuaris finals que accediran al sistema	2 usuaris	50	250
		Total	**7.230 euros**

Taula 2 Valoració econòmica pessimista

Així, l'escenari, tenint en compte la contractació de llicencies para al accés al sistema per part de cinc usuaris finals, serà en el cas més optimista en torn als 6.250 euros, mentre que en el pessimista estarà en torn als 7.250 euros; el que suposa una variació, per a un nombre similar d'usuaris, de uns 1.000 euros.

[2] Document complert del conveni col·lectiu nacional d'empresas d'ingeniería i oficines d'estudis técnics: http://www.comfia.net/archivos/XVCONVENIOINGENIERIASDefinitivo_Comfia2.pdf

1.7. Documents a entregar

Els documents a entregar en el TFC son els següents:

Pla de Treball: Planificació i estimació de les activitats necessàries per a dur a terme els objectius previstos. Entrega el 20/03/2011 a la bústia del consultor.

Memòria: Document final del projecte, la memòria no deuria superar les 60 planes. Entrega el 12/06/2011 a les bústies de cadascun dels membres del Tribunal d'Avaluació i a la bústia del consultor. El missatge te de contenir un resum del treball d'unes 200 paraules.

Presentació virtual: Resum clar i concís del treball realitzat i dels resultats obtinguts. Entrega el 12/06/2011 a les bústies de cadascun dels membres del Tribunal d'Avaluació i a la bústia del consultor.

A continuació es detallen que s'entregarà a cada PAC del TFC per a realitzar la avaluació continuada.

1.7.1. PAC1

Aquest pla de treball s'entrega com si fos una primera versió inacabada de la memòria del projecte. El contingut d'aquesta primera versió de la memòria procurarà els següents punts de la memòria:

- 1 Introducció.
- 1.1 Descripció del TFC.
- 1.2 Objectius generals i específics.
- 1.3 Enfocament i mètode seguit.
- 1.4 Rols i participants.
- 1.5 Planificació del projecte.
- 1.5.1 Descomposició estructural de les activitats.
- 1.5.2 Planificació amb fites i temporització.
- 1.5.3 Diagrama de Gantt.
- 1.5.4 Anàlisis de riscos.

- 1.6 Valoració econòmica.
- 1.7 Documents a entregar.
- 1.7.1 PAC1.
- 1.7.2 PAC2.
- 1.7.3 PAC3.
- 1.7.4 Producte final.

L'entrega de la PAC1 es realitza el dia 20/03/2011.

1.7.2. PAC2

Aquesta entrega engloba principalment tot el referent al disseny de la aplicació. Es revisen els requeriments proporcionats per l'enunciat per a la BBDD, e iniciem el disseny.

Aquesta PAC serà recollida principalment en els punts:

- 2. Disseny.
- 3. Implementació.

Els punts que es volen tractar i entregar en aquesta PAC son principalment:

- La elaboració del disseny conceptual de la BBDD mitjançant diagrames E/R.
- Instal·lació i configuració del SGBD Oracle.
- Elaboració del disseny físic de la BBDD i la construcció a partir dels scripts de creació de BBDD

L'entrega de la PAC2 es realitza el dia 17/04/2011.

1.7.3. PAC3

Aquesta entrega engloba principalment tot el referent a la implementació i proves internes de la aplicació.

Aquesta PAC serà recollirà principalment en els punts:

- 3. Implementació.

- 4. Testing.

Els punts que es volen tractar i entregar en aquesta PAC son principalment:

- Construïm i creem les seqüencies y disparadors, els procediments emmagatzemats i les funcions necessàries.

- Finalment, revisem el codi desenvolupat amb l'objectiu de tindre-ho tot en un únic arxiu.sql

L'entrega de la PAC3 es realitza el dia 15/05/2011.

1.7.4. Producte final

La entrega final del TFC.

En aquesta última entrega s'entregarà tot el projecte:

- Producte final: La aplicació de gestió de sancions i amonestacions.

- Memòria: Aquest mateix document. La memòria sintetitza el treball realitzat. Mostra la informació rellevant que permeti entendre el problema plantejat por el TFC, la metodologia utilitzada per a la seva resolució i mostrar la resolució del problema plantejat.

- Presentació: Es un document de síntesis de 20 transparències com màxim que deu sintetitzar de forma clara el TFC.

L'entrega final es realitza el dia 12/06/2011 i el tribunal virtual es farà des de el 20/06/2011 fins al 26/06/2011.

2. Disseny

Primerament, identificarem les entitats i relacions necessàries per al desenvolupament de la base de dades. Amb la informació recollida i organitzada realitzarem el disseny lògic de la base de dades.

2.1. Disseny de la BBDD

De la llista de requisits del TFC podem extreure la que hi han diversos tipus de taules depenent de la seva funció.

El primer tipus son els tipus de entitats que contenen els valors dels mestres en si, es a dir, taules d'alumnes, de professors, ... El segon tipus ens mostra les taules que ens defineixen una relació entre dues o més entitats de mestres. Per exemple, AlumneAssignatura ens mostraria quines assignatures te un alumne. L'últim tipus son taules de referencia o auxiliars, aquestes entitats no estan referenciades a res en concret, però s'utilitzen com referencia de dades, ja siguin per a consultar o contrastar.

Després de revisar els requeriments del projecte dividim les taules necessàries entre els tres tipus abans esmentats:

Taula	Informació	Tipus
Instituts	Dades sobre els diferents instituts a gestionar.	Mestre
Assignatures	Dades de les diferents assignatures. Al ser el pla d'estudis comú per a tots els instituts les assignatures no pertanyen a un institut en concret.	Mestre
Professors	Dades sobre els professors. Es defineix que un professor només pot pertànyer a un institut.	Mestre
Cursos	Dades dels cursos dels instituts. Un curs representa una classe, un conjunt d'alumnes, amb el seu professor responsable.	Mestre
Alumnes	Dades dels diferents alumnes. Un alumne pertany a un curs en concret.	Mestre
Sancions	Dades de les diferents definicions de sancions. Cada institut pot definir les seves pròpies sancions.	Mestre
Amonestacions	Dades de les diferents definicions d'amonestacions. Cada institut te la capacitat de definir les seves pròpies amonestacions.	Mestre

Taula 3 Entitats mestres

Taula	Informació	Tipus
AssignaturesProfessors	Guarda les dades de la relació d'assignatures en que un professor dona classe. Una assignatura pot estar donada per n professors, i un professor pot donar n assignatures.	Relació
AssignaturesCursos	Guarda les dades de la relació d'assignatures que hi ha en un curs donat. Una assignatura pot estar donada en n cursos, i un curs pot tindre n assignatures.	Relació
SancionsAlumne	Guarda les sancions que te un alumne.	Relació
AmonestacionsAlumne	Guarda les amonestacions que te un alumne.	Relació

Taula 4 Entitats de relació

Taula	Informació	Tipus
Calendari	Guarda les dades que defineixen el calendari de les assignatures.	Auxiliar
DiesFestius	Guarda els dies festius del any.	Auxiliar
HoresAtencio	Guarda les hores d'atenció a alumnes i pares que un professor por definir.	Auxiliar
Estadistiques	Guarda tot el conjunt de dades estadístiques que s'han precalculat.	Auxiliar
Logs	Guarda les dades amb els logs de l'aplicació	Auxiliar

Taula 5 Entitats auxiliars

De les entitats que s'han definit anteriorment podem extraure la següent informació en quant a establiment de relacions entre elles.

Taula A	Tipus de relació	Taula B
Institut	1:N	Alumnes
Institut	1:N	Cursos
Institut	1:N	Professors
Institut	1:N	Amonestacions
Institut	1:N	Sancions
Alumnes	1:N	AmonestacionsAlumne
Alumnes	1:N	SancionsAlumne
Cursos	1:N	Alumnes
Cursos	1:N	AmonestacionsAlumne
Cursos	1:N	SancionsAlumne
Cursos	1:N	AssignaturesCursos
Cursos	1:N	Calendari
Assignatures	1:N	AssignaturesCursos
Assignatures	1:N	AssignaturesProfessors

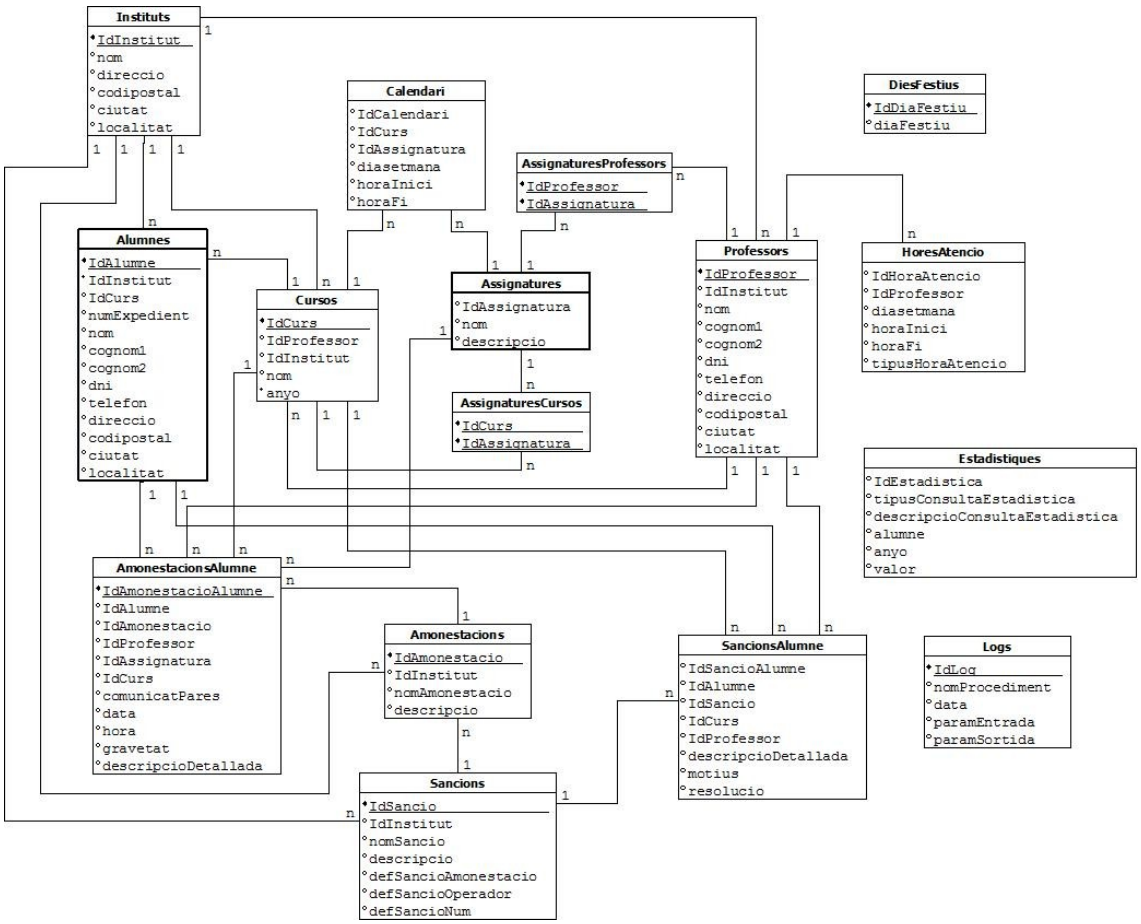
Assignatures	1:N	AmonestacionsAlumne
Assignatures	1:N	Calendari
Professors	1:N	Cursos
Professors	1:N	AssignaturesProfessors
Professors	1:N	AmonestacionsAlumne
Professors	1:N	SancionsAlumne
Professors	1:N	HoresAtencio
Amonestacions	1:N	AmonestacionsAlumne
Sancions	1:N	SancionsAlumne
Sancions	1:N	Amonestacions

Taula 6 Relacions entre les diferents taules

A partir del resultat de les definicions anteriors, es procedeix a transformar-lo en un model lògic relacional a partir de les següents regles:

- Les entitats originen relacions.
- Les interrelacions binaries 1:1 y 1:N originen claus foranies.

A continuació es mostra el diagrama lògic relacional resultant. El disseny lògic relacional contempla les diferents entitats i les seves relacions:



2.2. Disseny de la gestió dels Logs

Les crides realitzades als procediments emmagatzemats deixaran sempre constància a la taula de LOGS. La taula de LOGS també podrà ser utilitzada per a inserir registres provinents de l'execució d'algun disparador, per a que l'usuari pugui veure el que passa en el sistema.

La gestió de canvis realitzats en l'aplicació queden enregistrats en la taula de LOGS. La informació enregistrada es:

- El nom del procés que s'ha executat.
- L'hora a la que s'ha inserit la línia.
- Paràmetres d'entrada.
- Paràmetres de sortida.

Si la línia de la taula de LOGS fa referencia a una entrada creada per l'acció d'un disparador el paràmetre d'entrada mostrarà el "Id" del registre de la taula que ha llençat el disparador, i al camp de paràmetres de sortida mostrarà informació suficient per a facilitar la lectura al usuari.

La taula de LOGS es crea mitjançant aquesta sentencia:

```
CREATE TABLE LOGS
(
 IdLog        NUMBER              CONSTRAINT PK_LOGS PRIMARY KEY,
 nom          VARCHAR2(25 CHAR)   NOT NULL,
 data         DATE                NOT NULL,
 paramEntrada VARCHAR2(255 CHAR),
 paramSortida VARCHAR2(255 CHAR)
);

CREATE SEQUENCE s_Log INCREMENT BY 1 START WITH 1;

CREATE OR REPLACE
TRIGGER add_IdLog_LOGS
BEFORE INSERT ON LOGS
FOR EACH ROW
BEGIN
  SELECT s_Log.NEXTVAL INTO :NEW.IdLog FROM DUAL;
END add_IdLog_LOGS;
```

2.3. Disseny dels procediments d'ABM

Els requeriments del enunciat en lo referent a procediments emmagatzemats son els següents:

Requeriments per als procediments emmagatzemats

[...]
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.

[...]
Per a estandarditzar el sistema que s'ha de fer, es demana explícitament que els procediments emmagatzemats compleixin les condicions següents […]:

- Com a mínim disposaran d'un paràmetre de sortida anomenat RSP, de tipus string, que indicarà si l'execució ha finalitzat amb èxit (valor 'OK') o si ha fracassat (valor 'ERROR+TIPUS D'ERROR')
- Disposaran de tractament d'excepcions.
- Emmagatzemaran totes les crides a procediments que es facin en una taula de log, emmagatzemant el procediment executat, els paràmetres d'entrada i els de sortida.

Taula 7 Requeriments dels procediments

Prenent com a referencia els punts anteriors, i tenint en compte que els procediments tindran gestió d'excepcions, el cos dels procediments tindran la següent estructura:

```
PROCEDURE
(
 param1 (entrada o salida) tipo,
 […]
 RST OUT VARCHAR2
)
IS
BEGIN
[…]
COMMIT;
EXCEPTION
[…]
ROLLBAK;
END;
```

Per identificar els procediments emmagatzemats necessaris per a la base de dades, elaborem una taula que creua les entitats definides com a mestres de la base de dades amb els tres tipus generals d'operacions: Alta, Baixa, Modificació.

Això ens permet identificar una sèrie de procediments emmagatzemats que considerem imprescindibles per a el correcte funcionament de la aplicació. Alguns d'aquests procediments es consideren com millores desitjables que es desenvoluparan si els temps a del projecte ho permet.

Taula	Procediment	Tipus
Instituts	proc_InstitutAlta	Alta
Instituts	proc_InstitutBaixa	Baixa
Instituts	proc_InstitutModif	Modificació
Assignatures	proc_AssignaturaAlta	Alta
Assignatures	proc_AssignaturaBaixa	Baixa
Assignatures	proc_AssignaturaModif	Modificació
Professors	proc_ProfessorAlta	Alta
Professors	proc_ProfessorBaixa	Baixa
Professors	proc_ProfessorModif	Modificació
Cursos	proc_CursAlta	Alta
Cursos	proc_CursBaixa	Baixa
Cursos	proc_CursModif	Modificació
Alumnes	proc_AlumneAlta	Alta
Alumnes	proc_AlumneBaixa	Baixa
Alumnes	proc_AlumneModif	Modificació
Sancions	proc_SancioAlta	Alta
Sancions	proc_SancioBaixa	Baixa
Sancions	proc_SancioModif	Modificació

Amonestacions	proc_AmonestacioAlta	Alta
Amonestacions	proc_AmonestacioBaixa	Baixa
Amonestacions	proc_AmonestacioModif	Modificació
SancionsAlumne	proc_SancioAlumneAlta	Alta
SancionsAlumne	proc_SancioAlumneBaixa	Baixa
SancionsAlumne	proc_SancioAlumneModif	Modificació
AmonestacionsAlumne	proc_AmonesAlumneAlta	Alta
AmonestacionsAlumne	proc_AmonesAlumneBaixa	Baixa
AmonestacionsAlumne	proc_AmonesAlumneModif	Modificació

Taula 8 Llistat de procediments

Aquest procediments, no obstant, no representen una llista de procediments complerta. Un cop començat el desenvolupament de l'aplicació molt probablement es detectin necessitats de nous procediments o consultes més especialitzades.

2.4. Disseny del mòdul estadístic

El mòdul estadístic es una part molt important del treball final de carrera. La seva finalitat es l'accés a la informació resumida del sistema. El mòdul estadístic s'encarrega de tindre aquesta informació preparada. Tota la informació actualitzada del mòdul estadístic es troba en la taula ESTADISTIQUES.

Aquesta informació s'actualitza quan es produeix una nova inserció, modificació, o esborrat de un registre que afecti als càlculs. Així dons, esta pensada per a que amb una SELECT simple es pugui obtenir la dada actualitzada del tipus de estadística que volem per a un any concret, sense interferir en el rendiment de l'aplicació.

Els procediments necessaris per a realitzar l'actualització de les dades estan pensats per a ser cridats des de una única transacció. Aquests procediments seran el motor central del manteniment de tot el mòdul estadístic, i s'encarregaran de realitzar les diferents accions segons la taula d'estadístiques en totes les línies necessàries. La decisió d'implementar tota la gestió sobre una sèrie de crides als diferents procediments des de un únic procediment té a veure amb la propietat d'atomicitat de les transaccions que ens assegura que si una de les actualitzacions és incorrecta l'estat de la BBDD restaria íntegrament sense modificar, incloent el procediment que hagués fet la crida d' aquest últim.

L' estructura d' aquests procediments queda per tant dividida en les següents accions:

- Actualització/Inserció del valor "Numero d'amonestacions per alumne".
- Actualització/Inserció del valor "Numero de sancions per alumne i curs".

- Actualització/Inserció del valor “Mitjana d’amonestacions per professor i curs”.
- Actualització/Inserció del valor “Número de sancions per curs”.
- Actualització/Inserció del valor “Nom del alumne mes sancionat per curs”.
- Actualització/Inserció del valor “Nom del professor mes amonestador per curs”.
- Actualització/Inserció del valor “Mitjana de sancions per curs”.
- Actualització/Inserció del valor “Número d’alumnes sense amonestacions”.

Els diferents procediments tenen de ser disparats des de un únic procediment per a garantir la transacció, tot i això, no es necessari disparar tots els procediments sempre, per tant optimitzarem la crida a aquests tenint en compte les necessitats de ser disparats.

Mostrem en el quadre següent els moments en els que tenim de llençar cada estadística

Estadística	**Procediment de recàlcul**	**Actualitzar en...**
Numero d’amonestacions per alumne.	proc_estad_NumAmonesAlumne	ABM de AmonestacionsAlumnes
Numero de sancions per alumne i curs.	proc_estad_NumSancioAluCurs	ABM de SancionsAlumnes
Mitjana d’amonestacions per professor i curs.	proc_estad_MitjaAmonesProfCurs	ABM de AmonestacionsAlumnes
Número de sancions per curs.	proc_estad_NumSancionsCurs	ABM de SancionsAlumnes
Nom del alumne mes sancionat per curs.	proc_estad_AluMesSancionatCurs	ABM de SancionsAlumnes
Nom del professor mes amonestador per curs.	proc_estad_ProfMesAmonesCurs	ABM de AmonestacionsAlumnes
Mitjana de sancions per curs.	proc_estad_MitjaSancionsCurs	ABM de SancionsAlumnes
Número d’alumnes sense amonestacions.	proc_estad_NumAluSenseAmones	ABM de AmonestacionsAlumnes ABM de Alumnes

2.5. Disseny de la gestió d'amonestacions i sancions

La gestió d'amonestacions es realitza a partir dels procediments emmagatzemats de ABM de AmonestacionsAlumne.

El disseny d'aquest apartat es complica amb les sancions. Hi ha dos tipus definits de sancions:

- Sancions inserides manualment.
- Sancions inserides automàticament.

Les sancions inserides manualment es realitzen a partir del procediments emmagatzemats de ABM de SancionsAlumne. Les sancions automàtiques es creen tenint en compte les regles definides pels usuaris en la taula Sancions.

Al donar d'alta una amonestació es comprovarà a la taula sanció si hi ha alguna sanció definida que compleixi els requisits donats d'alta. Aquest procés te de ser dinàmic, per tant es treballarà amb un procediment que recorri els valors definits per al institut en qüestió, i llençarà les sancions necessàries quan es compleixin.

L'esquelet principal del procediment constarà d'un *cursor* on, per a cada sanció definida, s'utilitzarà un *execute immediate* de la consulta creada amb els valors propis de cada sanció. A continuació es pot observar part del codi utilitzat.

```
FOR cRecordset IN cCursor LOOP
  vEXECUTE := 'SELECT Count(IdAmonestacio) FROM AmonestacionsAlumne
                  WHERE IdAmonestacio = '|| cRecordset.DefSancioAmonestacio  || ' and
                  IdAlumne = ' || vIdAlumne || ' Group By IdAlumne Having count(*)'||
                 cRecordset.DefSancioOperador || cRecordset.DefSancioNum;
  BEGIN
    EXECUTE IMMEDIATE vEXECUTE INTO vValor;

    --CODI QUE S'ENCARREGA DE LA INSERCIÓ EN SANCIONSALUMNE

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
    NULL;
  END;
END LOOP;
```

2.6. Disseny dels procediments de consulta

Els requeriments del enunciat en lo referent a procediments emmagatzemats de consulta son els següents:

Requeriments per als procediments emmagatzemats
• 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.

Per a cada una de les consultes necessàries es crearà un procediment emmagatzemant, el procediment s'encarregarà de, tot rebent un parametre, tornar els valors bàsics que s'especifiquen en la consulta.

La sortida de les consultes es formatjarà per a que el resultat sigui visible de forma comprensible.

A continuació es llisten les consultes, els procediments que les criden, i els paràmetres necessaris per a la seva execució.

Consulta	**Procediment de la consulta**	**Paràmetres**
Llistat de totes les amonestacions imposades indicant-ne la seva informació bàsica.	proc_LlistatAmonestacions	IdInstitut
Llistat de tots els Alumnes d'un curs indicant-ne la seva informació bàsica.	proc_LlistatAlumnes	vIdCurs
Llistat de tots els tipus d'amonestacions i sancions disponibles per a aplicar en el institut.	proc_LlistatAmonesSancioInsti	IdInstitut
Llistat de totes les amonestacions i sancions d'un alumne.	proc_LlistatAmonesSancioAlu	vIdAlumne

3. Implementació

La construcció de la base de dades a partir del disseny lògic i la programació dels procediments emmagatzemats es realitza amb Oracle 10g. Aquesta aplicació ha permès la creació i definició de las sentencies que creen les taules de la base de dades, i la programació i proves dels procediments emmagatzemats.

Cada un dels elements construïts s'ha provat; las taules mitjançant insercions de dades per a comprovar el compliment de restriccions, i els procediments emmagatzemats mitjançant execucions sistemàtiques que han comprovat el seu comportament amb dades d'entrada tant correctes com incorrectes. En cas d'error s'ha verificat que es generaren les excepcions adequades i queden enregistrades a la taula de LOGS.

3.1. Creació de les taules

Seguint el disseny de les entitats esmentades al punt 2.1, es creen les taules.

Els scripts de creació de les taules es troben dins de l'arxiu *Create_Tables.sql*, adjuntat a la solució dins la subcarpeta ".\SQL\Desglossat SQL\"

Els tipus i mida dels camps de les taules s'han escollit tenint en compte cada cas en particular (Per exemple, el DNI seria un camp de tipus CHAR, ja que es de longitud fixa, i un camp de data de naixement seria de tipus DATE). Els camps que son de caràcter imprescindible per a la aplicació deuen ser emplenats obligatòriament, per tant es marquen en el moment de la creació de la taula com a camps amb la propietat NOT NULL. Es interesant destacar la utilització de clàusules CHECK en camps que nomes poden prendre uns determinats valors. La gran majoria de les taules tenen un camp de identificació únic, el qual es un valor autonumèric creat a partir d'un trigger i una seqüencia.

Una mostra de creació de taula (I els seu trigger per al camp autonumèric) es mostra a continuació. La resta de scripts es poden veure a l'arxiu *Create_Tables.sql.*

```
CREATE TABLE SANCIONS
(
 IdSancio NUMBER CONSTRAINT PK_SANCIONS PRIMARY KEY,
 IdInstitut CONSTRAINT FK_SANCIONS REFERENCES INSTITUTS(IdInstitut),
 nomSancio VARCHAR2(25 CHAR),
 descripcio VARCHAR2(255 CHAR),
 defSancioAmonestacio NUMBER,
 defSancioOperador VARCHAR2(1 CHAR),
 defSancioNum NUMBER,
 CHECK ((defSancioOperador = ''>'') OR (defSancioOperador = ''='') OR (defSancioOperador = ''<''))
);
```

```
CREATE SEQUENCE s_Sancio INCREMENT BY 1 START WITH 1;

CREATE OR REPLACE TRIGGER add_IdSancio_SANCIONS
BEFORE INSERT ON SANCIONS
FOR EACH ROW
BEGIN
  SELECT s_Sancio.NEXTVAL INTO :NEW.IdSancio FROM DUAL;
END add_IdSancio_SANCIONS;
```

3.2. Creació dels procediments d'ABM

Els noms utilitzats per als procediments emmagatzemats es suficientment descriptiu de la seva funció. Una descripció ampliada del propòsit de cada procediment i de la seva forma d'utilització pot trobar-se en la capçalera del codi font, en un format estandarditzat.

Els procediments emmagatzemats de Alta, Baixa i Modificació es troben dins l'arxiu *Create_Procedures_ABM.sql*, adjuntat a la solució dins la subcarpeta ".\SQL\Desglossat SQL\"

Tots els procediments emmagatzemats intenten tindre la mateixa estructura, primer es comprova si es compleixen totes les validacions necessàries abans d'executar-se, i en cas contrari s'aixeca una excepció. Tant si el procediment acaba correctament com si finalitza amb error, es guarda una entrada en la taula de log. Per a facilitar la gestió dels missatges a guardar en Log tots utilitzen variables per a emmagatzemar el nom del procediment i els paràmetres utilitzats, facilitant la gestió al desenvolupador si s'ha de modificar el procediment.

A continuació es mostra una mostra dels procediment emmagatzemats utilitzats per a l'Alta, Baixa i Modificació de una de les entitats principals. La resta de scripts es poden veure a l'arxiu *Create_Procedures_AMB.sql.*

```
CREATE OR REPLACE PROCEDURE proc_InstitutAlta
(
 vnom IN VARCHAR2,
 vdireccio IN VARCHAR2,
 vcodipostal IN VARCHAR2,
 vciutat IN VARCHAR2,
 vlocalitat IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);

BEGIN
```

```
vProcedimiento := ''proc_InstitutAlta'';
vParametres :=  vnom || '','' || vdireccio || '','' || vcodipostal || '','' || vciutat || '','' || vlocalitat;

IF (vnom IS NULL OR vnom = '') THEN RAISE debe_rellenarse; END IF;

   INSERT INTO Instituts (nom, direccio, codipostal, ciutat, localitat)
   VALUES (vnom, vdireccio, vcodipostal, vciutat, vlocalitat);

RSP := 'OK';

   INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
   VALUES (vProcedimiento, SYSDATE, vParametres, RSP);

EXCEPTION
WHEN STORAGE_ERROR THEN
   RSP := 'ERROR: No s´ha pogut insertar';
   INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
   VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
WHEN debe_rellenarse THEN
   RSP := 'ERROR: No s´han emplenat els paràmetres necessaris';
   INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
   VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
WHEN OTHERS THEN
   RSP := 'ERROR: Error genéric al donar d´alta';
   INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
   VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
END;
```

3.3. Creació dels procediments per a la gestió estadística

Els procediments emmagatzemats per a la gestió estadística tenen de ser transparents per al usuari, de totes formes, tot i no ser necessària la seva utilització fora de crides internes de la pròpia aplicació, els noms utilitzats per als procediments emmagatzemats son suficientment descriptius en quant a la seva funció i els paràmetres necessaris per a utilitzar-los. Una descripció ampliada del propòsit de cada procediment i de la seva forma d'utilització pot trobar-se en la capçalera del codi font, en un format estandarditzat. La informació sobre el disseny es pot trobar al punt 2.4 d'aquesta mateixa memòria.

Els procediments emmagatzemats relacionats amb la gestió estadística es troben dins l'arxiu *Create_Procedures_Estadistiques.sql*, adjuntat a la solució dins la subcarpeta ".\SQL\Desglossat SQL\". S'adjunta un dels procediments estadístics a continuació per a mostrar l'estructura i el mètode seguit.

```
CREATE OR REPLACE PROCEDURE proc_estad_NumAmonesAlumne
(
 vIdAlumne IN NUMBER
)
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
   BEGIN
     SELECT alumne INTO vValor from Estadistiques
     WHERE tipusConsultaEstadistica = 'NumAmonesAlumne' and alumne = vIdAlumne;

   EXCEPTION
    WHEN NO_DATA_FOUND THEN
    INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica,
                                        alumne, valor)
    VALUES ('NumAmonesAlumne', 'Numero d´amonestacions per alumne', vIdAlumne, 0);
   END;

 --Actualitzem el registre que pertoca
   SELECT count(*) INTO vValor FROM AmonestacionsAlumne WHERE idalumne = vIdAlumne;
   UPDATE Estadistiques SET valor = vValor
   WHERE tipusConsultaEstadistica = 'NumAmonesAlumne' AND alumne = vIdAlumne;
END;
```

3.4. Creació dels procediments per a la gestió d'amonestacions i sancions

Per a la gestió d'amonestacions i sancions es requeriran procediments emmagatzemats. S'ha comprovat la possibilitat de utilitzar triggers o vistes materialitzades, però per temes de eficiència i gestió de cascades en els triggers (que poden donar problemes de taules mutants) s'ha decidit aprofitar que, ja que l'alta de les línies de AmonestacionsAlumne es tenen de realitzar des de un procediment emmagatzemat de ABM, aprofitar aquest procediment per a llençar els recalculs necessaris en aquell moment. Tot i ser scripts interns de l'aplicació, els noms utilitzats per als procediments emmagatzemats son suficientment descriptius en quant a la seva funció i als paràmetres necessaris per a utilitzar-los. Una descripció ampliada de la forma de treball del procés automàtic de sancions es pot trobar al punt 2.5 de la memòria.

Els procediments emmagatzemats relacionats amb la gestió d'amonestacions i sancions es troben dins l'arxiu *Create_Procedures_AmonestacionsSancions.sql*, adjuntat a la solució dins la subcarpeta ".\SQL\Desglossat SQL\"

A continuació es mostra un dels procediment emmagatzemats utilitzats per a la gestió de sancions automàtiques. La resta de scripts, incloent els necessaris per a l'alta baixa i modificació de les amonestacions i sancions, es poden veure a dins de l'arxiu *Create_Procedures_AmonestacionsSancions.sql.*

```
CREATE OR REPLACE PROCEDURE proc_SancionsAutomatiques
(
 vIdAlumne IN NUMBER
)
IS

vCheck NUMBER;
vIdCurs NUMBER;
vEXECUTE VARCHAR2(4000);

CURSOR cCursor IS
  SELECT DefSancioAmonestacio, DefSancioOperador, DefSancioNum, idSancio, descripcio
  FROM Sancions
  WHERE IdInstitut IN (SELECT IdInstitut FROM ALUMNES WHERE IdAlumne = vIdAlumne);
BEGIN
  SELECT idCurs INTO vIdCurs FROM ALUMNES WHERE IdAlumne = vIdAlumne;
  FOR cRecordset IN cCursor LOOP
    vEXECUTE := 'SELECT Count(IdAmonestacio)
                  FROM AmonestacionsAlumne
                  WHERE IdAmonestacio = '||    cRecordset.DefSancioAmonestacio  || ' and
                          IdAlumne = ' || vIdAlumne || ' Group By IdAlumne Having count(*)'||
                          cRecordset.DefSancioOperador || cRecordset.DefSancioNum;

    BEGIN
      EXECUTE IMMEDIATE vEXECUTE INTO vCheck;
      proc_SancioAlumneAlta (vIdAlumne, cRecordset.idsancio, vIdCurs, NULL, cRecordset.descripcio,
                                  'Sanció Automàtica', NULL, vEXECUTE);

    EXCEPTION
      WHEN NO_DATA_FOUND THEN
      NULL;
    END;
  END LOOP;
END;
```

3.5. Creació dels procediments de consultes

Els procediments de consultes estan codificats de forma que els noms utilitzats siguin suficientment descriptius en quant a la seva funció i als paràmetres necessaris per a utilitzar-los. Una descripció ampliada de les consultes disponibles i la forma de devolució i formateig dels llistats de dades es pot trobar al punt 2.6 de la memòria.

Els procediments emmagatzemats relacionats amb el procediments de consulta es troben dins l'arxiu *Create_Procedures_Consultes.sql*, adjuntat a la solució dins la subcarpeta ".\SQL\"

La sortida de les consultes es formateja per a que el resultat sigui visible de una forma comprensible. A continuació es mostra un dels procediment emmagatzemats utilitzats per a consultes. Els altres procediments de scripts de consulta es poden veure a dins de l'arxiu *Create_Procedures_Consultes.sql.*

```
CREATE OR REPLACE PROCEDURE proc_LlistatAlumnes (vIdCurs IN NUMBER)
IS
vCurs VARCHAR2(25);
CURSOR cCursor IS
 select *
 from alumnes
 where idcurs = vIdCurs;
BEGIN
 SELECT nom into vCurs from Cursos where idcurs = vIdCurs;
 DBMS_OUTPUT.PUT_LINE(''Llistat d´alumnes del curs ''||vCurs );
 DBMS_OUTPUT.PUT_LINE('''');
 DBMS_OUTPUT.PUT_LINE(RPAD(''Nom'',26)||RPAD(''Primer cognom'',26)||
                      RPAD(''Segon cognom'',26)||RPAD(''DNI'',13));

 FOR cRecordset IN cCursor LOOP
    DBMS_OUTPUT.PUT_LINE(RPAD(cRecordset.nom,26)||RPAD(cRecordset.cognom1,26)||
                         RPAD(cRecordset.cognom2,26)||cRecordset.dni);
 END LOOP;
END;
```

4. Testing

4.1. Carrega inicial de dades

S'adjunta a l'aplicació un script que inserta dades de demostració. Aquest script s'encarrega d'esborrar la BBDD e inserta els valors de Demo. Un cop executat l'script podrem executar qualsevol prova que l'usuari vulgui amb les noves dades. Aquesta càrrega de dades es troba a l'arxiu *BBDD_Insert_Demo.sql*, adjuntat a la solució dins la subcarpeta ".\SQL\"

4.2. Creació del joc de proves de dades

Les proves consistiran en introduir dades i comprovar que es compleixen les restriccions especificades per els requeriments. Les dades no tenen perquè ser realistes, sino orientades a realitzar les proves. Per exemple, els cursos dels alumnes tindran 4 o 5 alumnes, en comptes dels 20 a 40 habituals.

S'ha de tindre en compte que les proves dissenyades no comproven exhaustivament totes i cadascunes de les funcionalitats del sistema (casos d'error possibles, casos particulars…) sino que s'ha tractat de fer que fos una prova representativa de les funcionalitats mes importants al temps que s'han provocat alguns errors controlats per a mostrar el comportament vers ells.

La seqüencia de proves planificada consistirà en el següent:

4.2.1. Operacions prèvies

Eliminar totes les dades de les taules. D'aquesta manera eliminem possibles resultats que puguin haver de proves anteriors. L'esborrat es realitza en l'ordre adequat per evitar problemes de restriccions entre taules. Aquest esborrat s'executa automàticament al executar el primer script de test.

Per a comprovar el correcte funcionament de la parella seqüencia-disparador de cada taula, hem fet alguns INSERT simples sobre cada taula comprovant el resultat amb una SELECT en la finestra d'execució SQL.

El test de la funcionalitat s'ha realitzat comprovant que els processos s'executen correctament així com verificant que les excepcions es compleixen quan els valors no son correctes o no processen les dades d'acord amb la lògica de negoci definida en el disseny.

En la carpeta del producte ".\SQL\Llançament TESTS\" trobarem els scripts preparats per ser executats en un entorn SQL d'Oracle. Aquest *scripts* estan preparats per comprovar el comentat en la anterior introducció. Els scripts emplenen la base de dades en alguns casos per

provar el test i al final de cada test si s'escau es realitza una inserció massiva de dades de cara a tenir dades per als següents tests.

4.2.2. Proves dels procediments de Alta, Baixa i Modificació

Aquest test es pot executar des de l'script:

".\SQL\Llançament TESTS\01_Test_Procediment_ABM.sql"

Les proves realitzades en aquest script s'encarregaran de inserir dades a partir dels procediments emmagatzemats ABM, també es modificaran i s'esborraran dades existents. Les dades tractades i el llançament dels procediments testejaran una mostra dels diferents missatges d'error que l'aplicació retorna. S'adjunta una taula amb la resolució del test.

Els procediments testejats son una mostra dels procediments existents, tots ells fets tenint en compte les possibles excepcions que es podrien donar en un entorn de treball normal.

Les dades inserides en aquesta part del test s'utilitzen més endavant per a continuar testejant l'aplicació. Després de les crides realitzades per al test es realitzen crides per a acabar d'emplenar les dades.

Procediment	Descripció	Resultat esperat	Estatus
PROC_INSTITUTALTA	Creem un primer institut amb les dades correctes	Sense errors	PASSAT
PROC_INSTITUTALTA	Creem un segon institut amb les dades correctes	Sense errors	PASSAT
PROC_INSTITUTALTA	Donem d'alta un registre però passant com a paràmetre un valor no vàlid	Missatge d'error informant de que falten paràmetres	PASSAT
PROC_ASSIGNATURAALTA	Creem una primera assignatura amb les dades correctes	Sense errors	PASSAT
PROC_ASSIGNATURAALTA	Creem una segona assignatura amb les dades correctes	Sense errors	PASSAT
PROC_ASSIGNATURAMODIF	Modifiquem la assignatura amb Id=2 per a canviar-li la descripció	Sense errors	PASSAT
PROC_ASSIGNATURABAIXA	Després d'emplenar varies assignatures més esborrem la assignatura amb Id=10	Sense errors	PASSAT
PROC_PROFESSORALTA	Creem un professor amb les dades correctes	Sense errors	PASSAT
PROC_PROFESSORALTA	Intentem donar d'alta un registre però donant-li un valor per a Id de institut que no correspon amb cap institut existent a la BBDD	Missatge d'error informant del error al donar d'alta	PASSAT
PROC_CURSALTA	Intentem inserir un curs amb un	Missatge d'error informant de	PASSAT

	valor buit per a la descripció	que falten paràmetres	
PROC_ALUMNEALTA	Donem d'alta varis alumnes, tots ells amb les dades correctes	Sense errors	PASSAT
PROC_ALUMNEMODIF	Intentem modificar un alumne, però li donem dades no valides	Missatge d'error informant de que no s'han emplenat correctament totes les dades	PASSAT
PROC_ASSIGCURSOSALTA	Intentem inserir una relació de assignatures i cursos però no li posem valor al curs	Missatge d'error informant de que no s'han emplenat correctament totes les dades	PASSAT
PROC_ASSIGCURSOSALTA	Intentem donar d'alta una relació entre assignatures i cursos no existents	Missatge d'error informant del error al donar d'alta	PASSAT
PROC_ASSIGCURSOSMODIF	Intentem inserir una relació de assignatures i cursos però no li posem valor a la assignatura	Missatge d'error informant de que no s'han emplenat correctament totes les dades	PASSAT
PROC_ASSIGPROFALTA	Creem una associació de assignatura i professor amb les dades correctes	Sense errors	PASSAT
PROC_ASSIGPROFBAIXA	Eliminem una associació de assignatura professor però no introduïm els Id correctament	Missatge d'error informant de que no s'han emplenat correctament totes les dades	PASSAT
PROC_ASSIGPROFBAIXA	Eliminem una associació de assignatura professor	Sense errors	PASSAT
PROC_CALENDARIALTA	Creem una entrada en el calendari amb les dades correctes	Sense errors	PASSAT
PROC_CALENDARIALTA	Creem una entrada en el calendari amb dades que provocarien un solapament de hores	Missatge d'error informant que no es dona d'alta l'horari a causa del solapament	PASSAT
PROC_CALENDARIMODIF	Intentem modificar una entrada del calendari, però li donem valors no vàlids per a la modificació	Missatge d'error informant de que no s'han emplenat correctament totes les dades	PASSAT
PROC_DIESFESTIUSALTA	Creem un dia festiu amb les dades correctes	Sense errors	PASSAT
PROC_HORESATENCIOALTA	Creem una hora d'atenció amb les dades correctes	Sense errors	PASSAT
PROC_HORESATENCIOALTA	Creem una hora d'atenció amb dades que provocarien un solapament de hores	Missatge d'error informant que no es dona d'alta l'horari a causa del solapament	PASSAT

4.2.3. Proves de Amonestacions i Sancions

Per a executar aquest test abans es tindrà d'haver executat el test anterior, sino pot esser que no hi hagin dades necessàries per a l'execució correcta d'aquest script. Aquest test es pot executar des de l'script:

".\SQL\Llançament TESTS\02_Test_Procediment_AmonestacionsSancions.sql"

Les proves realitzades en aquest script s'encarregaran de inserir dades a partir dels procediments emmagatzemats per a la gestió de amonestacions i sancions, es comprovarà la creació automàtica de sancions, així com la creació manual d'aquestes. Les dades tractades i el llançament dels procediments testejaran una mostra dels diferents missatges d'error que l'aplicació retorna. S'adjunta una taula amb la resolució del test.

Els procediments testejats son una mostra dels procediments existents, tots ells fets tenint en compte les possibles excepcions que es podrien donar en un entorn de treball normal.

Les dades inserides en aquesta part del test s'utilitzen més endavant per a continuar testejant l'aplicació. Després de les crides realitzades per al test es realitzen crides per a acabar d'emplenar les dades.

Procediment	**Descripció**	**Resultat esperat**	**Estatus**
PROC_AMONESTACIOALTA	Inserim una definició de amonestació amb dades correctes	Sense errors	PASSAT
PROC_AMONESTACIOALTA	Inserim una definició de amonestació que referencia a un institut no existent	Missatge d'error informant del error al donar d'alta	PASSAT
PROC_AMONESTACIOMODIF	Es modifica una definició de amonestació, però no se li emplena el nou nom	Missatge d'error informant de que no s'han emplenat correctament totes les dades	PASSAT
PROC_SANCIOALTA	Inserim una definició de sanció amb dades correctes	Sense errors	PASSAT
PROC_SANCIOALTA	Inserim una definició de sanció però com a operador elegim un valor no compres entre <, = , ó >	Missatge d'error informant del error al donar d'alta	PASSAT
PROC_AMONESALUMNEALTA	Inserim una amonestació per a un alumne amb dades correctes	Sense errors	PASSAT
PROC_AMONESALUMNEMODIF	Es modifica una amonestació d'un alumne però li posem una referencia a curs no vàlida	Missatge d'error informant del error al donar d'alta	PASSAT
PROC_SANCIOALUMNEALTA	Donem d'alta una sanció manual a un alumne	Sense errors	PASSAT
PROC_SANCIONSAUTOMATIQUES	Aquest procediment es té d'haver executat automàticament ell sol	Comprovem que existeixen dues línies en la taula SancionsAlumne (la automàtica i la creada manual)	PASSAT

4.2.4. Proves de Estadístiques

Aquest test es pot executar des de l'script:

".\SQL\Llançament TESTS\03_Test_Procediment_Estadistic.sql"

Les proves realitzades en aquest script s'encarregaran de verificar que les dades inserides al llarg dels tests realitzats han actualitzat correctament les estadístiques. S'adjunta una taula amb la resolució del test.

Procediment	Descripció	Resultat esperat	Estatus
PROC_AMONESALUMNEALTA	Donem d'alta varies amonestacions més per a veure resultats en les taules d'estadístiques	Sense errors	PASSAT
PROC_ESTAD_NUMALUSENSEAMONES	Mirem el valor de la estadística	Esperem com a resultat un 14	PASSAT
PROC_ESTAD_NUMAMONESALUMNE	Mirem el valor de la estadística per a l'alumne amb Id=1	Esperem com a resultat un 5	PASSAT
PROC_ESTAD_MITJAAMONESPROFCURS	Mirem el valor de la estadística per al curs amb Id=1	Esperem com a resultat un 3	PASSAT
PROC_ESTAD_PROFMESAMONESCURS	Mirem el valor de la estadística per al curs amb Id=1	Esperem com a resultat el professor A	PASSAT
PROC_ESTAD_NUMSANCIOALUCURS	Mirem el valor de la estadística per al curs amb Id=1 i l'alumne amb Id=1	Esperem com a resultat un 1	PASSAT
PROC_ESTAD_NUMSANCIONSCURS	Mirem el valor de la estadística per al curs amb Id=1	Esperem com a resultat un 2	PASSAT
PROC_ESTAD_ALUMESSANCIONATCURS	Mirem el valor de la estadística per al curs amb Id=1	Esperem com a resultat l'alumne 2	PASSAT
PROC_ESTAD_MITJASANCIONSCURS	Mirem el valor de la estadística	Esperem com a resultat un 2	PASSAT

4.2.5. Comprovació de les consultes

Un cop acabades les insercions produïdes durant les proves realitzades en els tests anteriors, comprovarem a traves dels procediments de consulta si ens coincideixen les dades amb el que ens deurien donar. S'adjunten captures de pantalla dels resultats de cada consulta.

Hem inserit quatre alumnes pertanyents al curs amb Id=1. Llencem el nostre procediment de consulta per a veure els alumnes del primer curs:

PROC_LLISTATALUMNES(1);

```
Llistat d´alumnes del curs Curs 1

Nom                    Primer cognom          Segon cognom           DNI
AlumNom 1              AlumCognom1 1          AlumCognom2 1          01234567A
AlumNom 2              AlumCognom1 2          AlumCognom2 2          01234567B
AlumNom 3              AlumCognom1 3          AlumCognom2 3          01234567C
AlumNom 4              AlumCognom1 4          AlumCognom2 4          01234567D
```

Realitzem la validació del segon dels procediments de consulta. Ens te de donar un llistat de les amonestacions i sancions corresponents al alumne amb Id=1.

PROC_LLISTATAMONESSANCIOALU(1);

Tipus	Nom	Descripcio	Data
Amonestacio	alumne_arriba_tard	Arriba tard	13/10/11
Amonestacio	alumne_arriba_tard	Arriba tard	12/10/11
Amonestacio	alumne_arriba_tard	Arriba tard	11/10/11
Amonestacio	alumne_arriba_tard	Arriba tard	10/10/11
Amonestacio	alumne_soroll	Sorollos	12/10/11
Sancio	Sancio_HoraExtra	Quedar-se una hora extra d	

El següent procediment de consulta ens mostra les diferents amonestacions i sancions donades de alta per a un institut donat. Mostrem a continuació el llistat retornat per el procediment:

PROC_LLISTATAMONESSANCIOINSTI(1);

Llistat de tipus d´amonestacions i sancions del insitut IES Joan Miró

Tipus	Nom	Descripcio
Amonestacio	alumne_arriba_tard	Quan l´alumne arriba tard a classe.
Amonestacio	alumne_soroll	L´alumne fa molt de soroll en classe.
Amonestacio	alumne_malparlat	L´alumne és malparlat
Amonestacio	alumne_no_deures	L´alumne no fa els deures.
Sancio	Sancio_HoraExtra	Quedar-se una hora extra d´estudi durant una setmana
Sancio	Sancio_AvaluacioExtra	Prova d´avaluació extra

Per últim, comprovem el procediment que ens mostra totes les amonestacions imposades en el institut amb Id=1.

PROC_LLISTATAMONESTACIONS(1);

Llistat d´amonestacions imposades en l´institut IES Joan Miró

Amonestacio	IdAlumne	Data
alumne_arriba_tard	1	10/10/11
alumne_arriba_tard	1	11/10/11
alumne_arriba_tard	1	12/10/11
alumne_arriba_tard	1	13/10/11
alumne_soroll	1	12/10/11
alumne_soroll	2	10/10/11
alumne_soroll	2	11/10/11
alumne_soroll	2	12/10/11
alumne_soroll	2	12/10/11

Com podem observar totes les consultes retornen els valors esperats.

4.2.6. Comprovació dels Logs

Comprovarem que en la taula de LOGS s'han inserit correctament les crides als diferents procediments emmagatzemats, i que es mostrin les sortides corresponents. Tenim de tindre en compte que al arxiu de Log tenim de veure-hi reflexades també les crides automàtiques per a les sancions automàtiques.

Amb una consulta ràpida podem veure els registres emmagatzemats a la taula Log en ordre descendent. Comprovem que els valors que ens mostra la taula Log son els esperats. Es mostren a continuació els últims 50 valors

SELECT * FROM LOGS ORDER BY IdLog DESC

IDLOG	NOM	DATA	PARAMENTRADA	PARAMSORTIDA
138	proc_AmonesAlumneAlta	11/06/11	2,3,3,4,1,Sí,12/10/11,7,2,Sorollos	OK
137	proc_AmonesAlumneAlta	11/06/11	2,3,2,3,1,Sí,12/10/11,7,3,Sorollos	OK
136	proc_AmonesAlumneAlta	11/06/11	2,3,2,3,1,Sí,11/10/11,6,2,Sorollos	OK
135	proc_AmonesAlumneAlta	11/06/11	2,3,1,3,1,Sí,10/10/11,6,2,Sorollos	OK
134	proc_AmonesAlumneAlta	11/06/11	1,3,3,4,1,Sí,12/10/11,6,2,Sorollos	OK
133	proc_AmonesAlumneAlta	11/06/11	1,1,1,2,1,Sí,13/10/11,8,4,Arriba tard	OK
132	proc_SancioAlumneAlta	11/06/11	2,,1,1,Pegar a un company de classe,Pegar a un company de classe,Dues setmanes expulsat de l´escola	OK
131	proc_AmonesAlumneAlta	11/06/11	1,1,1,2,1,Sí,12/10/11,8,4,Arriba tard	OK
130	proc_SancioAlumneAlta	11/06/11	1,1,1,,Quedar-se una hora extra d´estudi durant una setmana,Sanció Automàtica,	OK
129	proc_AmonesAlumneAlta	11/06/11	1,1,1,2,1,Sí,11/10/11,8,2,Arriba tard	OK
128	proc_AmonesAlumneModif	11/06/11	1,1,1,4,15,1,Sí,10/10/11,8,2,Arriba tard	ERROR: Error genèric al modificar
127	proc_AmonesAlumneAlta	11/06/11	1,1,1,2,1,Sí,10/10/11,8,2,Arriba tard	OK
126	proc_SancioAlta	11/06/11	2,Sancio_AvaluacioExtra,Prova d´avaluació extra	OK
125	proc_SancioAlta	11/06/11	2,Sancio_HoraExtra,Quedar-se una hora extra d´estudi durant una setmana	OK
124	proc_SancioAlta	11/06/11	1,Sancio_AvaluacioExtra,Prova d´avaluació extra	OK
123	proc_SancioAlta	11/06/11	1,Sancio_HoraExtra,Quedar-se una hora extra d´estudi durant una setmana - amb ?	ERROR: Error genèric al donar d´alta
122	proc_SancioAlta	11/06/11	1,Sancio_HoraExtra,Quedar-se una hora extra d´estudi durant una setmana	OK
121	proc_AmonestacioAlta	11/06/11	2,alumne_no_deures,L´alumne no fa els deures.	OK
120	proc_AmonestacioAlta	11/06/11	2,alumne_malparlat,L´alumne és malparlat	OK
119	proc_AmonestacioAlta	11/06/11	2,alumne_soroll,L´alumne fa molt de soroll en classe.	OK
118	proc_AmonestacioAlta	11/06/11	2,alumne_arriba_tard,Quan l´alumne arriba tard a classe.	OK
117	proc_AmonestacioAlta	11/06/11	1,alumne_no_deures,L´alumne no fa els deures.	OK
116	proc_AmonestacioAlta	11/06/11	1,alumne_malparlat,L´alumne és malparlat	OK
115	proc_AmonestacioAlta	11/06/11	1,alumne_soroll,L´alumne fa molt de soroll en classe.	OK
114	proc_AmonestacioModif	11/06/11	1,1,,Sense nom	ERROR: No s´han emplenat els paràmetres necessaris
113	proc_AmonestacioAlta	11/06/11	3,alumne_arriba_tard,Quan l´alumne arriba tard a classe.	ERROR: Error genèric al donar d´alta
112	proc_AmonestacioAlta	11/06/11	1,alumne_arriba_tard,Quan l´alumne arriba tard a classe.	OK
111	proc_HoresAtencioAlta	11/06/11	2,3,6,7,Hora de consulta	OK
110	proc_HoresAtencioAlta	11/06/11	2,1,6,7,Hora d´atencio	OK
109	proc_HoresAtencioAlta	11/06/11	1,4,6,7,Hora de consulta	OK
108	proc_HoresAtencioAlta	11/06/11	1,2,6,7,Hora d´atencio	ERROR: La data a inserir provocaria un solapament de hores
107	proc_HoresAtencioAlta	11/06/11	1,2,6,7,Hora d´atencio	OK
106	proc_DiesFestiusAlta	11/06/11	24/12/11	OK
105	proc_DiesFestiusAlta	11/06/11	11/09/11	OK
104	proc_DiesFestiusAlta	11/06/11	15/08/11	OK
103	proc_DiesFestiusAlta	11/06/11	01/05/11	OK
102	proc_DiesFestiusAlta	11/06/11	05/01/11	OK
101	proc_CalendariAlta	11/06/11	1,7,5,5,6	OK
100	proc_CalendariAlta	11/06/11	1,7,4,5,6	OK
99	proc_CalendariAlta	11/06/11	1,7,3,5,6	OK
98	proc_CalendariAlta	11/06/11	1,6,2,4,5	OK
97	proc_CalendariAlta	11/06/11	1,6,1,5,6	OK
96	proc_CalendariAlta	11/06/11	1,5,5,4,5	OK
95	proc_CalendariAlta	11/06/11	1,5,1,4,5	OK
94	proc_CalendariAlta	11/06/11	1,4,4,3,4	OK
93	proc_CalendariAlta	11/06/11	1,4,3,4,5	OK
92	proc_CalendariAlta	11/06/11	1,4,2,3,4	OK
91	proc_CalendariAlta	11/06/11	1,3,5,2,4	OK
90	proc_CalendariAlta	11/06/11	1,3,3,2,4	OK
89	proc_CalendariAlta	11/06/11	1,3,1,2,4	OK

5. Conclusions

Un cop finalitzat tot el procés de desenvolupament d'acord amb la planificació realitzada, podem verificar que els objectius que vam identificar al inici i que es detallaven en el enunciat del TFC s'han acomplert. La base de dades, i els procediments emmagatzemats han sigut dissenyats i construïts d'acord amb els requeriments especificats i s'entreguen acompanyant aquest document.

Segons el que s'ha vist durant el temps que ha durat el desenvolupament del projecte, es pot extreure una sèrie de conclusions relacionades no només amb el que es sol·licitava en ell, sino també genèriques sobre la planificació de projectes, el cicle de vida dels mateixos, la metodologia a seguir i la convinença del us dels SGBD comercials.

En primer lloc, hem pogut comprovar com resulta essencial realitzar una bona planificació del projecte. Aquesta planificació a de ser realista, i en aquest sentit es important contemplar períodes festius i altres absències previsibles, per a que no impactin en la temporització. A més a més, a l'hora de planificar, convé dedicar un temps suficient per a les probes de la aplicació i també per a les correccions necessàries. Si no es dimensionen bé aquests temps, es pot córrer el risc de que s'aproximin les dades d'entrega i el desenvolupament no estigui funcionant correctament.

Per a poder corregir els errors el més ràpid possible, de forma que afectin el menys possible en etapes posteriors del projecte i que hi hagi la major quantitat de temps disponible per a arreglar-los, s'imposa una estratègia de detecció d'errors. Aquesta estratègia, al llarg del cicle de vida del projecte, s'ha realitzat mitjançant successives entregues de documentació, les quals ha anat revisant i validant el client, senyalant els errors, imprecisions o divergències de interpretació que s'han produït en cada etapa. En aquest sentit, es primordial que existeixi un compromís no sòls per part del desenvolupador, sino també per part del client, de compliment de fites. Si bé el desenvolupador té de realitzar les seves entregues en las dades acordades, s'ha pogut comprovar com la labor del client ha resultat imprescindible, i com una ràpida lectura de la documentació i resposta a la mateixa ha permès que el projecte no vagi arrastrant errors des de las primeres fases fins les últimes, moment en el qual seria molt més difícil corregir els errors.

Per la banda tècnica, les bases de dades estan molt vinculades a la feina que tinc actualment, per tant, mai està de mes aprendre noves tecnologies o refrescar coneixements.

Conclusió, una etapa molt interessant de formació, que espero que tanqui un cicle de la meva vida de molt esforç, 4 anys treballant i estudiant, amb els horaris que ofereix actualment el mon laboral de la branca de tecnologia, moltes vegades amb sobrecarrega d'hores.

6. Glossari

Backup: Copia de seguretat.

Base de Dades: Es un conjunt estructurat de dades que representa, entre altres, entitats i les seves interrelacions, amb integració i compartimentació de dades.

BBDD: Veure Base de Dades.

Clau forana: En les BBDD es parla d' una clau forana quan un atribut d' una taula referència a un atribut d' una altra taula.

Clau primària: Es defineix en les bases de dades com l' atribut que identifica una entitat i que pren exactament un valor únic per a cada ocurrència. En el nostre projecte hi ha exemples molt clars com per exemple la llicència d' un jugador.

Consistència: La consistència és una propietat de les Bases de Dades relacionals que ens assegura que no existeixen contradiccions entre les dades i les seves propietats i relacions.

Disparador: Veure Trigger.

Disseny Conceptual: Etapa del disseny d'una base de dades que obté una estructura de la informació de la futura base de dades independentment de la tecnologia que es vulgui utilitzar.

Disseny Lògic: Etapa del disseny d'una base de dades que parteix del resultat del disseny conceptual y el transforma de forma que s'adapti al model del SGBD amb el que es desitja implementar la base de dades.

ETIG: Enginyeria Tècnica en Informàtica de Gestió.

Excepció: Les excepcions són les respostes controlades que dóna un programa quan s'introdueix un o uns valors d' entrada incorrectes o que no formen part del domini contemplat.

Interrelació: Associació entre entitats.

Log: Procés de seguiment d'execució de processos.

PAC: Prova d'Avaluació Continuada.

Plataforma: En el nostre cas parlem de plataforma per a referir-nos a tots els elements software necessaris i sobre el quals ha de funcionar l' SGBD que utilitzem.

PL/SQL: En el SGBD Oracle, PL/SQL es un llenguatge procedimental que estén SQL. El seu propòsit es combinar el llenguatge de la base de dades amb un llenguatge de programació procedimental per a la definició d'accions o funcions.

Procediment emmagatzemat: Acció o funció definida per un usuari que proporciona un determinat servei. Una vegada ha sigut creat, es guarda en la BBDD i passa a ser tractat com un objecte més de aquesta. La execució d'un procediment pot retornar ningun, un o més valors.

Script: Codi font que executa un procés.

SGBD: Veure Sistema de Gestió de Bases de Dades.

Sistema de Gestió de Bases de Dades: Software que gestiona y controla bases de dades. Les seves principals funcions son les de facilitar la utilització simultània a molts usuaris de tipus diferents, independitzar al usuari del mon físic i mantenir la integritat de los dades.

SQL: S*tructured Query Language*. Llenguatge pensat per a descriure, crear, actualitzar i consultar bases de dades. Actualment l'utilitzen casi tots els SGBD del mercat.

TFC: Treball Final de Carrera.

Trigger: Acció o procediment emmagatzemat que s'executa automàticament quan es realitza una operació de INSERT, DELETE o UPDATE sobre alguna taula de la BBDD.

7. Bibliografia

Material de Bases de Dades I - Jaume Sistac Planas (UOC)

Material de Bases de Dades II - Jaume Sistac Planas (UOC)

Material de Sistemes de gestió de Bases de Dades - Jaume Sistac Planas (UOC)

Material de Enginyeria del Software I - Benet Campderrich Falgueras (UOC)

http://es.wikipedia.org/wiki/Desarrollo_en_cascada

http://salarios.infojobs.net/resultados.cfm?sueldo=+bases+de+datos&o_id=2

http://www.comfia.net/archivos/XVCONVENIOINGENIERIASDefinitivo_Comfia2.pdf

http://www.techonthenet.com/oracle/index.php

http://www.psoug.org/library.html

http://www.oracle.com/technology/documentation/index.html

http://www.wikioracle.es/

http://www.infor.uva.es/~chernan/Bases/Teoria/TySQL.pdf

http://www.devjoker.com/contenidos/Tutorial-PLSQL/48/Excepciones-en-PLSQL.aspx

Annexos: Codi de l'aplicació

A continuació posem el codi utilitzat en l'aplicació:

Creació de Taules

```
BEGIN

--INSTITUTS
-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE INSTITUTS
(
 IdInstitut NUMBER CONSTRAINT PK_INSTITUTS PRIMARY KEY,
 nom VARCHAR2(25 CHAR),
 direccio VARCHAR2(25 CHAR),
 codipostal VARCHAR2(5 CHAR),
 ciutat VARCHAR2(25 CHAR),
 localitat VARCHAR2(25 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Institut INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdInstitut_INSTITUTS
BEFORE INSERT ON INSTITUTS
FOR EACH ROW
BEGIN
SELECT s_Institut.NEXTVAL INTO :NEW.IdInstitut
FROM DUAL;
END add_IdInstitut_INSTITUTS;';



--ASSIGNATURES
-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE ASSIGNATURES
(
 IdAssignatura NUMBER CONSTRAINT PK_ASIGNATURES PRIMARY KEY,
 nom VARCHAR2(25 CHAR),
 descripcio VARCHAR2(255 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Assignatura INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdAssignatura_ASSIGNATURES
BEFORE INSERT ON ASSIGNATURES
FOR EACH ROW
BEGIN
SELECT s_Assignatura.NEXTVAL INTO :NEW.IdAssignatura
FROM DUAL;
END add_IdAssignatura_ASSIGNATURES;';
```

```
--PROFESSORS
-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE PROFESSORS
(
 IdProfessor NUMBER CONSTRAINT PK_PROFESSORS PRIMARY KEY,
 IdInstitut CONSTRAINT FK_PROFESSORS REFERENCES INSTITUTS(IdInstitut),
 nom VARCHAR2(50 CHAR),
 cognom1 VARCHAR2(50 CHAR),
 cognom2 VARCHAR2(50 CHAR),
 dni VARCHAR2(9 CHAR),
 telefon VARCHAR2(25 CHAR),
 direccio VARCHAR2(25 CHAR),
 codipostal VARCHAR2(5 CHAR),
 ciutat VARCHAR2(25 CHAR),
 localitat VARCHAR2(25 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Professor INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdProfessor_PROFESSORS
BEFORE INSERT ON PROFESSORS
FOR EACH ROW
BEGIN
SELECT s_Professor.NEXTVAL INTO :NEW.IdProfessor
FROM DUAL;
END add_IdProfessor_PROFESSORS;';



--CURSOS
-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE CURSOS
(
 IdCurs NUMBER CONSTRAINT PK_CURSOS PRIMARY KEY,
 IdProfessor CONSTRAINT FK_1_CURSOS REFERENCES PROFESSORS(IdProfessor),
 IdInstitut CONSTRAINT FK_2_CURSOS REFERENCES INSTITUTS(IdInstitut),
 nom VARCHAR2(25 CHAR),
 anyo NUMBER
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Curs INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdCurs_CURSOS
BEFORE INSERT ON CURSOS
FOR EACH ROW
BEGIN
SELECT s_Curs.NEXTVAL INTO :NEW.IdCurs
FROM DUAL;
```

```
END add_IdCurs_CURSOS;';


--ALUMNES
-------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE ALUMNES
(
 IdAlumne NUMBER CONSTRAINT PK_ALUMNES PRIMARY KEY,
 IdInstitut CONSTRAINT FK_1_ALUMNES REFERENCES INSTITUTS(IdInstitut),
 IdCurs CONSTRAINT FK_2_ALUMNES REFERENCES CURSOS(IdCurs),
 numExpedient NUMBER,
 nom VARCHAR2(50 CHAR),
 cognom1 VARCHAR2(50 CHAR),
 cognom2 VARCHAR2(50 CHAR),
 dni VARCHAR2(9 CHAR),
 telefon VARCHAR2(25 CHAR),
 direccio VARCHAR2(50 CHAR),
 codipostal VARCHAR2(5 CHAR),
 ciutat VARCHAR2(25 CHAR),
 localitat VARCHAR2(25 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Alumne INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdAlumne_ALUMNES
BEFORE INSERT ON ALUMNES
FOR EACH ROW
BEGIN
SELECT s_Alumne.NEXTVAL INTO :NEW.IdAlumne
FROM DUAL;
END add_IdAlumne_ALUMNES;';



--ASSIGNATURESPROFESSORS
-------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE ASSIGNATURESPROFESSORS
(
 IdProfessor CONSTRAINT FK_1_ASSIGNATURESPROFESSORS REFERENCES PROFESSORS(IdProfessor),
 IdAssignatura CONSTRAINT FK_2_ASSIGNATURESPROFESSORS REFERENCES
ASSIGNATURES(IdAssignatura),
 CONSTRAINT PK_ASSIGNATURESPROFESSORS PRIMARY KEY(IdProfessor, IdAssignatura)
)';


--ASSIGNATURESCURSOS
-------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE ASSIGNATURESCURSOS
(
 IdCurs CONSTRAINT FK_1_ASSIGNATURESCURSOS REFERENCES CURSOS(IdCurs),
 IdAssignatura CONSTRAINT FK_2_ASSIGNATURESCURSOS REFERENCES ASSIGNATURES(IdAssignatura),
 CONSTRAINT PK_ASSIGNATURESCURSOS PRIMARY KEY(IdCurs, IdAssignatura)
)';
```

```
--CALENDARI
-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE CALENDARI
(
 IdCalendari NUMBER CONSTRAINT PK_CALENDARI PRIMARY KEY,
 IdCurs CONSTRAINT FK_1_CALENDARI REFERENCES CURSOS(IdCurs),
 IdAssignatura CONSTRAINT FK_2_CALENDARI REFERENCES ASSIGNATURES(IdAssignatura),
 diasetmana NUMBER,
 horaInici NUMBER,
 horaFi NUMBER,
 CHECK (horaInici<HoraFi)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Calendari INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdCalendari_CALENDARI
BEFORE INSERT ON CALENDARI
FOR EACH ROW
BEGIN
SELECT s_Calendari.NEXTVAL INTO :NEW.IdCalendari
FROM DUAL;
END add_IdCalendari_CALENDARI;';



--DIESFESTIUS
-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE DIESFESTIUS
(
 IdDiaFestiu NUMBER CONSTRAINT PK_DIESFESTIUS PRIMARY KEY,
 diaFestiu DATE
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_DiaFestiu INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdDiaFestiu_DIESFESTIUS
BEFORE INSERT ON DIESFESTIUS
FOR EACH ROW
BEGIN
SELECT s_DiaFestiu.NEXTVAL INTO :NEW.IdDiaFestiu
FROM DUAL;
END add_IdDiaFestiu_DIESFESTIUS;';



--HORESATENCIO
-----------------------------------------------------------------------
```

```
EXECUTE IMMEDIATE 'CREATE TABLE HORESATENCIO
(
 IdHoraAtencio NUMBER CONSTRAINT PK_HORESATENCIO PRIMARY KEY,
 IdProfessor CONSTRAINT FK_HORESATENCIO REFERENCES PROFESSORS(IdProfessor),
 diasetmana NUMBER,
 horaInici NUMBER,
 horaFi NUMBER,
 tipusHoraAtencio VARCHAR2(25 CHAR),
 CHECK (horaInici<HoraFi)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_HoraAtencio INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdHoraAtencio_HORESATENCIO
BEFORE INSERT ON HORESATENCIO
FOR EACH ROW
BEGIN
SELECT s_HoraAtencio.NEXTVAL INTO :NEW.IdHoraAtencio
FROM DUAL;
END add_IdHoraAtencio_HORESATENCIO;';



--ESTADISTIQUES
-------------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE ESTADISTIQUES
(
 IdEstadistica NUMBER CONSTRAINT PK_ESTADISTIQUES PRIMARY KEY,
 tipusConsultaEstadistica VARCHAR2(25 CHAR),
 descripcioConsultaEstadistica VARCHAR2(255 CHAR),
 alumne NUMBER,
 anyo NUMBER,
 valor VARCHAR2(255 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Estadistica INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdEstadistica_ESTADIS
BEFORE INSERT ON ESTADISTIQUES
FOR EACH ROW
BEGIN
SELECT s_Estadistica.NEXTVAL INTO :NEW.IdEstadistica
FROM DUAL;
END add_IdEstadistica_ESTADIS;';



--SANCIONS
-------------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE SANCIONS
(
 IdSancio NUMBER CONSTRAINT PK_SANCIONS PRIMARY KEY,
 IdInstitut CONSTRAINT FK_SANCIONS REFERENCES INSTITUTS(IdInstitut),
```

```
 nomSancio VARCHAR2(25 CHAR),
 descripcio VARCHAR2(255 CHAR),
 defSancioAmonestacio NUMBER,
 defSancioOperador VARCHAR2(1 CHAR),
 defSancioNum NUMBER,
 CHECK ((defSancioOperador = ''>'') OR (defSancioOperador = ''='') OR (defSancioOperador = ''<''))
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Sancio INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdSancio_SANCIONS
BEFORE INSERT ON SANCIONS
FOR EACH ROW
BEGIN
SELECT s_Sancio.NEXTVAL INTO :NEW.IdSancio
FROM DUAL;
END add_IdSancio_SANCIONS;';



--AMONESTACIONS
-------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE AMONESTACIONS
(
 IdAmonestacio NUMBER CONSTRAINT PK_AMONESTACIONS PRIMARY KEY,
 IdInstitut CONSTRAINT FK_1_AMONESTACIONS REFERENCES INSTITUTS(IdInstitut),
 nomAmonestacio VARCHAR2(25 CHAR),
 descripcio VARCHAR2(255 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Amonestacio INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdAmonestacio_AMONES
BEFORE INSERT ON AMONESTACIONS
FOR EACH ROW
BEGIN
SELECT s_Amonestacio.NEXTVAL INTO :NEW.IdAmonestacio
FROM DUAL;
END add_IdAmonestacio_AMONES;';



--SANCIONSALUMNE
-------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE SANCIONSALUMNE
(
 IdSancioAlumne NUMBER CONSTRAINT PK_SANCIONSALUMNE PRIMARY KEY,
 IdAlumne CONSTRAINT FK_1_SANCIONSALUMNE REFERENCES ALUMNES(IdAlumne),
 IdSancio CONSTRAINT FK_2_SANCIONSALUMNE REFERENCES SANCIONS(IdSancio),
 IdCurs CONSTRAINT FK_3_SANCIONSALUMNE REFERENCES CURSOS(IdCurs),
 IdProfessor CONSTRAINT FK_4_SANCIONSALUMNE REFERENCES PROFESSORS(IdProfessor),
```

```
 descripcioDetallada VARCHAR2(255 CHAR),
 motius VARCHAR2(255 CHAR),
 resolucio VARCHAR2(255 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_SancioAlumne INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdSancioAlumne_SANCIONSALU
BEFORE INSERT ON SANCIONSALUMNE
FOR EACH ROW
BEGIN
SELECT s_SancioAlumne.NEXTVAL INTO :NEW.IdSancioAlumne
FROM DUAL;
END add_IdSancioAlumne_SANCIONSALU;';



--AMONESTACIONSALUMNE
-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE TABLE AMONESTACIONSALUMNE
(
 IdAmonestacioAlumne NUMBER CONSTRAINT PK_AMONESTACIONSALUMNE PRIMARY KEY,
 IdAlumne CONSTRAINT FK_1_AMONESTACIONSALUMNE REFERENCES ALUMNES(IdAlumne),
 IdAmonestacio CONSTRAINT FK_2_AMONESTACIONSALUMNE REFERENCES
AMONESTACIONS(IdAmonestacio),
 IdProfessor CONSTRAINT FK_3_AMONESTACIONSALUMNE REFERENCES PROFESSORS(IdProfessor),
 IdAssignatura CONSTRAINT FK_4_AMONESTACIONSALUMNE REFERENCES
ASSIGNATURES(IdAssignatura),
 IdCurs CONSTRAINT FK_5_AMONESTACIONSALUMNE REFERENCES CURSOS(IdCurs),
 comunicatPares VARCHAR2(25 CHAR),
 data DATE,
 hora NUMBER,
 gravetat NUMBER,
 descripcioDetallada VARCHAR2(255 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_AmonestacioAlumne INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdAmonestacioAlu_AMONESALU
BEFORE INSERT ON AMONESTACIONSALUMNE
FOR EACH ROW
BEGIN
SELECT s_AmonestacioAlumne.NEXTVAL INTO :NEW.IdAmonestacioAlumne
FROM DUAL;
END add_IdAmonestacioAlu_AMONESALU;';



--LOGS
-----------------------------------------------------------------------
```

```
EXECUTE IMMEDIATE 'CREATE TABLE LOGS
(
 IdLog NUMBER CONSTRAINT PK_LOGS PRIMARY KEY,
 nom VARCHAR2(25 CHAR),
 data DATE,
 paramEntrada VARCHAR2(255 CHAR),
 paramSortida VARCHAR2(255 CHAR)
)';

EXECUTE IMMEDIATE 'CREATE SEQUENCE s_Log INCREMENT BY 1 START WITH 1';

EXECUTE IMMEDIATE 'CREATE OR REPLACE
TRIGGER add_IdLog_LOGS
BEFORE INSERT ON LOGS
FOR EACH ROW
BEGIN
SELECT s_Log.NEXTVAL INTO :NEW.IdLog
FROM DUAL;
END add_IdLog_LOGS;';

END;
/
```

Creació de procediments de Estadístiques

```
BEGIN

--Actualització/Inserció del valor “Numero d’amonestacions per alumne”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_NumAmonesAlumne
(
 vIdAlumne IN NUMBER
)
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
 BEGIN
   SELECT alumne INTO vValor from Estadistiques where tipusConsultaEstadistica =
''NumAmonesAlumne'' and alumne = vIdAlumne;
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
   INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, alumne, valor)
   VALUES (''NumAmonesAlumne'', ''Numero d´amonestacions per alumne'', vIdAlumne, 0);
 END;
 --Actualitzem el registre que pertoca
 SELECT count(*) INTO vValor FROM AmonestacionsAlumne WHERE idalumne = vIdAlumne;
 UPDATE Estadistiques SET valor = vValor WHERE tipusConsultaEstadistica = ''NumAmonesAlumne'' AND
alumne = vIdAlumne;
END;';


--Actualització/Inserció del valor “Numero de sancions per alumne i curs”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_NumSancioAluCurs
(
 vIdAlumne IN NUMBER,
 vIdCurs IN NUMBER
)
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
 BEGIN
   SELECT alumne INTO vValor from Estadistiques where tipusConsultaEstadistica =
''NumSancionsAlumneCurs'' and alumne = vIdAlumne and anyo = vIdCurs;
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
   INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, alumne, anyo,
valor)
   VALUES (''NumSancionsAlumneCurs'', ''Numero de sancions per alumne i curs'', vIdAlumne, vIdCurs,
0);
 END;
 --Actualitzem el registre que pertoca
 SELECT count(*) into vValor FROM SancionsAlumne WHERE idalumne = vIdAlumne and idcurs = vIdCurs;
 UPDATE Estadistiques SET valor = vValor
 WHERE tipusConsultaEstadistica = ''NumSancionsAlumneCurs'' AND alumne = vIdAlumne AND anyo =
vIdCurs;
```

```
END;';


--Actualització/Inserció del valor “Mitjana d’amonestacions per professor i curs”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_MitjaAmonesProfCurs
(
 vIdCurs IN NUMBER
)
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
 BEGIN
  SELECT anyo INTO vValor from Estadistiques where tipusConsultaEstadistica = ''MitjaAmonesProfCurs''
and anyo = vIdCurs;
 EXCEPTION
  WHEN NO_DATA_FOUND THEN
  INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, anyo, valor)
  VALUES (''MitjaAmonesProfCurs'', ''Mitjana d´amonestacions per professor i curs'', vIdCurs, 0);
 END;
 --Actualitzem el registre que pertoca
 SELECT ROUND(Avg(NumAmones),2) INTO vValor FROM (select count(*) NumAmones from
amonestacionsalumne where idcurs = vIdCurs group by idprofessor) SubSelect;
 UPDATE Estadistiques SET valor = vValor
 WHERE tipusConsultaEstadistica = ''MitjaAmonesProfCurs'' AND anyo = vIdCurs;
END;';


--Actualització/Inserció del valor “Número de sancions per curs”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_NumSancionsCurs
(
 vIdCurs IN NUMBER
)
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
 BEGIN
  SELECT anyo INTO vValor from Estadistiques where tipusConsultaEstadistica = ''NumSancionsCurs'' and
anyo = vIdCurs;
 EXCEPTION
  WHEN NO_DATA_FOUND THEN
  INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, anyo, valor)
  VALUES (''NumSancionsCurs'', ''Numero de sancions per curs'', vIdCurs, 0);
 END;
 --Actualitzem el registre que pertoca
 SELECT count(*) into vValor FROM SancionsAlumne WHERE idcurs = vIdCurs;
 UPDATE Estadistiques SET valor = vValor WHERE tipusConsultaEstadistica = ''NumSancionsCurs'' AND
anyo = vIdCurs;
END;';


--Actualització/Inserció del valor “Nom del alumne mes sancionat per curs”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_AluMesSancionatCurs
(
```

```
 vIdCurs IN NUMBER
)
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
 BEGIN
   SELECT anyo INTO vValor from Estadistiques where tipusConsultaEstadistica =
''AlumneMesSancionatCurs'' and anyo = vIdCurs;
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
   INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, anyo, valor)
   VALUES (''AlumneMesSancionatCurs'', ''Nom del alumne mes sancionat per curs'', vIdCurs, 0);
 END;
 --Actualitzem el registre que pertoca
 SELECT nom || '' '' || cognom1 || '' '' || cognom2 INTO vValor FROM (select idalumne from
sancionsalumne where idcurs = vIdCurs group by idalumne order by count(*) desc) SubSelect, Alumnes
where SubSelect.idalumne = alumnes.idalumne and rownum=1;
 UPDATE Estadistiques SET valor = vValor
 WHERE tipusConsultaEstadistica = ''AlumneMesSancionatCurs''AND anyo = vIdCurs;
END;';


--Actualització/Inserció del valor “Nom del professor mes amonestador per curs”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_ProfMesAmonesCurs
(
 vIdCurs IN NUMBER
)
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
 BEGIN
   SELECT anyo INTO vValor from Estadistiques where tipusConsultaEstadistica =
''ProfesorMesAmonesCurs'' and anyo = vIdCurs;
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
   INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, anyo, valor)
   VALUES (''ProfesorMesAmonesCurs'', ''Nom del professor mes amonestador per curs'', vIdCurs, 0);
 END;
 --Actualitzem el registre que pertoca
 SELECT nom || '' '' || cognom1 || '' '' || cognom2 INTO vValor FROM (select idprofessor, count(*) from
amonestacionsalumne where idcurs = vIdCurs group by idprofessor order by count(*) desc) SubSelect,
Professors where SubSelect.idprofessor = Professors.idprofessor and rownum=1;
 UPDATE Estadistiques SET valor = vValor
 WHERE tipusConsultaEstadistica = ''ProfesorMesAmonesCurs'' and anyo = vIdCurs;
END;';


--Actualització/Inserció del valor “Mitjana de sancions per curs”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_MitjaSancionsCurs
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
```

```
 BEGIN
   SELECT anyo INTO vValor from Estadistiques where tipusConsultaEstadistica = ''MitjaSancionsCurs'';
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
   INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, valor)
   VALUES (''MitjaSancionsCurs'', ''Mitjana de sancions per curs'', 0);
 END;
 --Actualitzem el registre que pertoca
 SELECT ROUND(Avg(NumSancio),2) INTO vValor FROM (select count(*) NumSancio from
sancionsalumne group by idcurs) SubSelect;
 UPDATE Estadistiques SET valor = vValor
 WHERE tipusConsultaEstadistica = ''MitjaSancionsCurs'';
END;';


--Actualització/Inserció del valor “Número d’alumnes sense amonestacions”.
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_estad_NumAluSenseAmones
IS
vValor VARCHAR(255 CHAR);
BEGIN
 --Comprobem si existeix la linea a actualizar en el modul estadistic, i si no existeix la creem
 BEGIN
   SELECT anyo INTO vValor from Estadistiques where tipusConsultaEstadistica =
''NumAlumnesSenseAmones'';
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
   INSERT INTO Estadistiques (tipusConsultaEstadistica, descripcioConsultaEstadistica, valor)
   VALUES (''NumAlumnesSenseAmones'', ''Número d´alumnes sense amonestacions'', 0);
 END;
 --Actualitzem el registre que pertoca
 SELECT count(*) INTO vValor FROM alumnes WHERE idalumne not in (SELECT DISTINCT idalumne FROM
AmonestacionsAlumne);
 UPDATE Estadistiques SET valor = vValor
 WHERE tipusConsultaEstadistica = ''NumAlumnesSenseAmones'';
END;';


END;
/
```

Creació de procediments de Consultes

```
BEGIN

EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_LlistatAmonestacions (vIdInstitut IN
NUMBER)
IS
vInstitut VARCHAR2(25);
CURSOR cCursor IS
  select amonestacions.nomamonestacio, amonestacionsalumne.idalumne, amonestacionsalumne.data
  from amonestacions, amonestacionsalumne
  where amonestacions.idamonestacio = amonestacionsalumne.idamonestacio
    and amonestacions.idinstitut = vIdInstitut;
BEGIN
  SELECT nom into vInstitut from instituts where idinstitut = vIdInstitut;
  DBMS_OUTPUT.PUT_LINE(''Llistat d´amonestacions imposades en l´institut ''||vInstitut );
  DBMS_OUTPUT.PUT_LINE('''');
  DBMS_OUTPUT.PUT_LINE(RPAD(''Amonestacio'',26)||RPAD(''IdAlumne'',16)||''Data'');
  FOR cRecordset IN cCursor LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(cRecordset.nomamonestacio,26)||RPAD(cRecordset.idalumne,16)||cR
ecordset.data);
  END LOOP;
END;';


EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_LlistatAlumnes (vIdCurs IN NUMBER)
IS
vCurs VARCHAR2(25);
CURSOR cCursor IS
  select *
  from alumnes
  where idcurs = vIdCurs;
BEGIN
  SELECT nom into vCurs from Cursos where idcurs = vIdCurs;
  DBMS_OUTPUT.PUT_LINE(''Llistat d´alumnes del curs ''||vCurs );
  DBMS_OUTPUT.PUT_LINE('''');
  DBMS_OUTPUT.PUT_LINE(RPAD(''Nom'',26)||RPAD(''Primer cognom'',26)||RPAD(''Segon
cognom'',26)||RPAD(''DNI'',13));
  FOR cRecordset IN cCursor LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(cRecordset.nom,26)||RPAD(cRecordset.cognom1,26)||RPAD(cRecords
et.cognom2,26)||cRecordset.dni);
  END LOOP;
END;';


EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_LlistatAmonesSancioInsti (vIdInstitut IN
NUMBER)
IS
vInstitut VARCHAR2(500);
CURSOR cCursor IS
  select  ''Amonestacio'' Tipus, nomAmonestacio Nom, Descripcio from amonestacions where
amonestacions.idinstitut = vIdInstitut
```

```
   UNION ALL
  select ''Sancio'' Tipus, nomSancio Nom, Descripcio from sancions where sancions.idinstitut =
vIdInstitut;
BEGIN
  SELECT nom into vInstitut from instituts where idinstitut = vIdInstitut;
  DBMS_OUTPUT.PUT_LINE(''Llistat de tipus d´amonestacions i sancions del insitut ''||vInstitut);
  DBMS_OUTPUT.PUT_LINE('''');
  DBMS_OUTPUT.PUT_LINE(RPAD(''Tipus'',26)||RPAD(''Nom'',26)||RPAD(''Descripcio'',26));
  FOR cRecordset IN cCursor LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(cRecordset.Tipus,26)||RPAD(cRecordset.Nom,26)||cRecordset.Descrip
cio);
  END LOOP;
END;';


EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_LlistatAmonesSancioAlu (vIdAlumne IN
NUMBER)
IS
vAlumne VARCHAR2(500);
CURSOR cCursor IS
  select  ''Amonestacio'' Tipus, nomAmonestacio Nom, descripciodetallada Descripcio, cast(data as
varchar2(10)) Data
  from amonestacionsalumne, amonestacions where amonestacions.idamonestacio =
amonestacionsalumne.idamonestacio and idalumne = vIdAlumne
    UNION ALL
  select ''Sancio'' Tipus, nomSancio Nom, descripciodetallada Descripcio, '''' as Data
  from sancionsalumne, sancions where sancions.idsancio = sancionsalumne.idsancio and idalumne =
vIdAlumne;
BEGIN
  select nom || '' '' || cognom1 || '' '' || cognom2 Nom into vAlumne from Alumnes where idalumne =
vIdAlumne;
  DBMS_OUTPUT.PUT_LINE(''Llistat d´amonestacions i sancions de l´alumne ''||vAlumne);
  DBMS_OUTPUT.PUT_LINE('''');
  DBMS_OUTPUT.PUT_LINE(RPAD(''Tipus'',26)||RPAD(''Nom'',26)||RPAD(''Descripcio'',26)||''Data'');
  FOR cRecordset IN cCursor LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(cRecordset.Tipus,26)||RPAD(cRecordset.Nom,26)||RPAD(cRecordset.
Descripcio,26)||cRecordset.data);
  END LOOP;
END;';
END;
/
```

Creació de procediments de Amonestacions i Sancions

```
BEGIN

--SANCIONS
--Alta---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_SancioAlta
(
 vidinstitut IN NUMBER,
 vnomSancio IN VARCHAR2,
 vdescripcio IN VARCHAR2,
 vdefSancioAmonestacio IN NUMBER,
 vdefSancioOperador IN VARCHAR2,
 vdefSancioNum IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_SancioAlta'';
vParametres := vidinstitut || '','' || vnomSancio || '','' || vdescripcio;

IF (vnomSancio IS NULL OR vnomSancio = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdefSancioAmonestacio IS NULL OR vdefSancioAmonestacio = '''') THEN RAISE debe_rellenarse; END
IF;
IF (vdefSancioOperador IS NULL OR vdefSancioOperador = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdefSancioNum IS NULL OR vdefSancioNum = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into SANCIONS (idinstitut, nomSancio, descripcio, defSancioAmonestacio, defSancioOperador,
defSancioNum)
VALUES (vidinstitut, vnomSancio, vdescripcio, vdefSancioAmonestacio, vdefSancioOperador,
vdefSancioNum);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa----------------------------------------------------------------------
```

```
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_SancioBaixa (vid IN VARCHAR2, RSP OUT
Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_SancioBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM SANCIONS WHERE idSancio = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';

--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_SancioModif
(
 vid IN NUMBER,
 vidinstitut IN NUMBER,
 vnomSancio IN VARCHAR2,
 vdescripcio IN VARCHAR2,
 vdefSancioAmonestacio IN NUMBER,
 vdefSancioOperador IN VARCHAR2,
 vdefSancioNum IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_SancioModif'';
vParametres :=  vid || '','' || vidinstitut || '','' || vnomSancio || '','' || vdescripcio;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vnomSancio IS NULL OR vnomSancio = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdefSancioAmonestacio IS NULL OR vdefSancioAmonestacio = '''') THEN RAISE debe_rellenarse; END
IF;
IF (vdefSancioOperador IS NULL OR vdefSancioOperador = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdefSancioNum IS NULL OR vdefSancioNum = '''') THEN RAISE debe_rellenarse; END IF;
```

```
UPDATE SANCIONS SET
idinstitut = vidinstitut,
nomSancio = vnomSancio,
descripcio = vdescripcio,
defSancioAmonestacio = vdefSancioAmonestacio,
defSancioOperador = vdefSancioOperador,
defSancioNum = vdefSancioNum
WHERE idsancio = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';


--AMONESTACIONS
--Alta-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AmonestacioAlta
(
 vidinstitut IN NUMBER,
 vnomAmonestacio IN VARCHAR2,
 vdescripcio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AmonestacioAlta'';
vParametres := vidinstitut || '','' || vnomAmonestacio || '','' || vdescripcio;

IF (vnomAmonestacio IS NULL OR vnomAmonestacio = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into AMONESTACIONS (idinstitut, nomAmonestacio, descripcio)
VALUES (vidinstitut, vnomAmonestacio, vdescripcio);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
```

```
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AmonestacioBaixa (vid IN VARCHAR2, RSP
OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_AmonestacioBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM AMONESTACIONS WHERE idAmonestacio = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';

--Modificacio-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AmonestacioModif
(
 vid IN NUMBER,
 vidinstitut IN NUMBER,
 vnomAmonestacio IN VARCHAR2,
 vdescripcio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
```

```
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AmonestacioModif'';
vParametres :=  vid || '','' || vidinstitut || '','' || vnomAmonestacio || '','' || vdescripcio;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vnomAmonestacio IS NULL OR vnomAmonestacio = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE AMONESTACIONS SET
idinstitut = vidinstitut,
nomAmonestacio = vnomAmonestacio,
descripcio = vdescripcio
WHERE idamonestacio = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';



--SANCIONSALUMNE
--Alta-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_SancioAlumneAlta
(
 vIdAlumne IN NUMBER,
 vIdSancio IN NUMBER,
 vIdCurs IN NUMBER,
 vIdProfessor IN NUMBER,
 vdescripcioDetallada IN VARCHAR2,
 vmotius IN VARCHAR2,
 vresolucio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_SancioAlumneAlta'';
vParametres := vIdAlumne || '','' || vIdSancio || '','' || vIdCurs || '','' || vIdProfessor || '','' ||
vdescripcioDetallada || '','' || vmotius || '','' || vresolucio;
```

```
Insert Into SANCIONSALUMNE (IdAlumne, IdSancio, IdCurs, IdProfessor, descripcioDetallada, motius,
resolucio)
VALUES (vIdAlumne, vIdSancio, vIdCurs, vIdProfessor, vdescripcioDetallada, vmotius, vresolucio);

--Executem la gestió de estadistiques
proc_estad_NumSancioAluCurs(vIdAlumne, vIdCurs);
proc_estad_NumSancionsCurs(vIdCurs);
proc_estad_AluMesSancionatCurs(vIdCurs);
proc_estad_MitjaSancionsCurs;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_SancioAlumneBaixa (vid IN VARCHAR2,
RSP OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_SancioAlumneBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM SANCIONSALUMNE WHERE idSancioAlumne = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';
```

```
--Modificacio----------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_SancioAlumneModif
(
 vid IN NUMBER,
 vIdAlumne IN NUMBER,
 vIdSancio IN NUMBER,
 vIdCurs IN NUMBER,
 vIdProfessor IN NUMBER,
 vdescripcioDetallada IN VARCHAR2,
 vmotius IN VARCHAR2,
 vresolucio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_SancioAlumneModif'';
vParametres :=  vid || '','' || vIdAlumne || '','' || vIdSancio || '','' || vIdCurs || '','' || vIdProfessor || '',''
|| vdescripcioDetallada || '','' || vmotius || '','' || vresolucio;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE SANCIONSALUMNE SET
IdAlumne = vIdAlumne,
IdSancio = vIdSancio,
IdCurs = vIdCurs,
IdProfessor = vIdProfessor,
descripcioDetallada = vdescripcioDetallada,
motius = vmotius,
resolucio = vresolucio
WHERE idsancioalumne = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';


--PROCEDIMENT INTERN per a les SancionsAutomatiques
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_SancionsAutomatiques
```

```
(
 vIdAlumne IN NUMBER
)
IS

vCheck NUMBER;
vIdCurs NUMBER;
vEXECUTE VARCHAR2(4000);

CURSOR cCursor IS
  SELECT DefSancioAmonestacio, DefSancioOperador, DefSancioNum, idSancio, descripcio
  FROM Sancions
  WHERE IdInstitut IN (SELECT IdInstitut FROM ALUMNES WHERE IdAlumne = vIdAlumne);
BEGIN
  SELECT idCurs INTO vIdCurs FROM ALUMNES WHERE IdAlumne = vIdAlumne;
  FOR cRecordset IN cCursor LOOP
    vEXECUTE := ''SELECT Count(IdAmonestacio) FROM AmonestacionsAlumne WHERE IdAmonestacio =
''||
              cRecordset.DefSancioAmonestacio  || '' and IdAlumne = '' || vIdAlumne || '' Group By
IdAlumne Having count(*)''||
              cRecordset.DefSancioOperador || cRecordset.DefSancioNum;
    BEGIN
      EXECUTE IMMEDIATE vEXECUTE INTO vCheck;
      proc_SancioAlumneAlta (vIdAlumne, cRecordset.idsancio, vIdCurs, NULL, cRecordset.descripcio,
''Sanció Automàtica'', NULL, vEXECUTE);
    EXCEPTION
      WHEN NO_DATA_FOUND THEN
      NULL;
    END;
  END LOOP;
END;';



--AMONESTACIONSALUMNE
--Alta-----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AmonesAlumneAlta
(
 vIdAlumne NUMBER,
 vIdAmonestacio NUMBER,
 vIdProfessor NUMBER,
 vIdAssignatura NUMBER,
 vIdCurs NUMBER,
 vcomunicatPares VARCHAR2,
 vdata DATE,
 vhora NUMBER,
 vgravetat NUMBER,
 vdescripcioDetallada VARCHAR2,
 RSP OUT Varchar2
)
IS
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AmonesAlumneAlta'';
```

```
vParametres := vIdAlumne || ''," || vIdAmonestacio || ","  || vIdProfessor || ","  || vIdAssignatura || ","
|| vIdCurs || ","  || vcomunicatPares || ","  || vdata || ","  || vhora || ","  || vgravetat || ","  ||
vdescripcioDetallada;

Insert Into AMONESTACIONSALUMNE (IdAlumne, IdAmonestacio, IdProfessor, IdAssignatura, IdCurs,
comunicatPares, data, hora, gravetat, descripcioDetallada)
VALUES (vIdAlumne, vIdAmonestacio, vIdProfessor, vIdAssignatura, vIdCurs, vcomunicatPares, vdata,
vhora, vgravetat, vdescripcioDetallada);

--Executem la comprobació de sancions automàtiques i estadistiques
proc_estad_NumAmonesAlumne(vIdAlumne);
proc_estad_MitjaAmonesProfCurs(vIdCurs);
proc_estad_ProfMesAmonesCurs(vIdCurs);
proc_estad_NumAluSenseAmones;
proc_SancionsAutomatiques(vIdAlumne);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AmonesAlumneBaixa (vid IN VARCHAR2,
RSP OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_AmonesAlumneBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM AMONESTACIONSALUMNE WHERE  IdAmonestacioAlumne = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
```

```
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';

--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AmonesAlumneModif
(
 vid IN NUMBER,
 vIdAlumne NUMBER,
 vIdAmonestacio NUMBER,
 vIdProfessor NUMBER,
 vIdAssignatura NUMBER,
 vIdCurs NUMBER,
 vcomunicatPares VARCHAR2,
 vdata DATE,
 vhora NUMBER,
 vgravetat NUMBER,
 vdescripcioDetallada VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AmonesAlumneModif'';
vParametres :=  vid || '','' || vIdAlumne || '','' || vIdAmonestacio || '','' || vIdProfessor || '','' ||
vIdAssignatura || '','' || vIdCurs || '','' || vcomunicatPares || '','' || vdata || '','' || vhora || '','' ||
vgravetat || '','' || vdescripcioDetallada;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE AMONESTACIONSALUMNE SET
IdAlumne = vIdAlumne,
IdAmonestacio = vIdAmonestacio,
IdProfessor = vIdProfessor,
IdAssignatura = vIdAssignatura,
IdCurs = vIdCurs,
comunicatPares = vcomunicatPares,
data = vdata,
hora = vhora,
gravetat = vgravetat,
descripcioDetallada = vdescripcioDetallada
WHERE IdAmonestacioAlumne = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
```

```
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';


END;
/
```

Creació de procediments de Alta, Baixa i Modificació

```
BEGIN

--INSTITUTS
--Alta-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_InstitutAlta
(
 vnom IN VARCHAR2,
 vdireccio IN VARCHAR2,
 vcodipostal IN VARCHAR2,
 vciutat IN VARCHAR2,
 vlocalitat IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_InstitutAlta'';
vParametres :=  vnom || '','' || vdireccio || '','' || vcodipostal || '','' || vciutat || '','' || vlocalitat;

IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;
Insert Into instituts (nom, direccio, codipostal, ciutat, localitat) VALUES (vnom, vdireccio, vcodipostal,
vciutat, vlocalitat);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_InstitutBaixa (vid IN VARCHAR2, RSP OUT
Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_InstitutBaixa'';
```

```
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM instituts WHERE idInstitut = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';


--Modificacio----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_InstitutModif
(
 vid IN NUMBER,
 vnom IN VARCHAR2,
 vdireccio IN VARCHAR2,
 vcodipostal IN VARCHAR2,
 vciutat IN VARCHAR2,
 vlocalitat IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_InstitutModif'';
vParametres :=  vid || '','' || vnom || '','' || vdireccio || '','' || vcodipostal || '','' || vciutat || '','' ||
vlocalitat;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;
UPDATE INSTITUTS SET
nom = vnom,
direccio = vdireccio,
codipostal = vcodipostal,
ciutat = vciutat,
localitat = vlocalitat
WHERE idinstitut = vid;

RSP := ''OK'';
```

```
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';



--ALUMNES
--Alta---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AlumneAlta
(
 vidinstitut IN NUMBER,
 vidcurs IN NUMBER,
 vnumexpedient IN VARCHAR2,
 vnom IN VARCHAR2,
 vcognom1 IN VARCHAR2,
 vcognom2 IN VARCHAR2,
 vdni IN VARCHAR2,
 vtelefon IN VARCHAR2,
 vdireccio IN VARCHAR2,
 vcodipostal IN VARCHAR2,
 vciutat IN VARCHAR2,
 vlocalitat IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AlumneAlta'';
vParametres := vidinstitut || '','' || vidcurs || '','' || vnumexpedient || '','' || vnom || '','' || vcognom1
|| '','' || vcognom2 || '','' || vdni;

IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;
IF (vcognom1 IS NULL OR vcognom1 = '''') THEN RAISE debe_rellenarse; END IF;
IF (vcognom2 IS NULL OR vcognom2 = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into ALUMNES (idinstitut, idcurs, numexpedient, nom, cognom1, cognom2, dni, telefon, direccio,
codipostal, ciutat, localitat)
VALUES (vidinstitut, vidcurs, vnumexpedient, vnom, vcognom1, vcognom2, vdni, vtelefon, vdireccio,
vcodipostal, vciutat, vlocalitat);
```

```
--Calculem estadístiques relacionades
proc_estad_NumAluSenseAmones;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AlumneBaixa (vid IN VARCHAR2, RSP OUT
Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_AlumneBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM ALUMNES WHERE idAlumne = vid;

--Calculem estadístiques relacionades
proc_estad_NumAluSenseAmones;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';

--Modificacio---------------------------------------------------------------
```

```
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AlumneModif
(
 vid IN NUMBER,
 vidinstitut IN NUMBER,
 vidcurs IN NUMBER,
 vnumexpedient IN VARCHAR2,
 vnom IN VARCHAR2,
 vcognom1 IN VARCHAR2,
 vcognom2 IN VARCHAR2,
 vdni IN VARCHAR2,
 vtelefon IN VARCHAR2,
 vdireccio IN VARCHAR2,
 vcodipostal IN VARCHAR2,
 vciutat IN VARCHAR2,
 vlocalitat IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AlumneModif'';
vParametres :=  vid || '','' || vidinstitut || '','' || vidcurs || '','' || vnumexpedient || '','' || vnom || '','' 
|| vcognom1 || '','' || vcognom2 || '','' || vdni;
vParametres := vParametres || '','' || vtelefon || '','' || vdireccio || '','' || vcodipostal || '','' || vciutat 
|| '','' || vlocalitat;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;
UPDATE ALUMNES SET
idinstitut = vidinstitut,
idcurs = vidcurs,
numexpedient = vnumexpedient,
nom = vnom,
cognom1 = vcognom1,
cognom2 = vcognom2,
dni = vdni,
telefon = vtelefon,
direccio = vdireccio,
codipostal = vcodipostal,
ciutat = vciutat,
localitat = vlocalitat
WHERE idalumne = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, 
vParametres, RSP);
WHEN debe_rellenarse THEN
```

```
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';



--PROFESSORS
--Alta---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_ProfessorAlta
(
 vidinstitut IN NUMBER,
 vnom IN VARCHAR2,
 vcognom1 IN VARCHAR2,
 vcognom2 IN VARCHAR2,
 vdni IN VARCHAR2,
 vtelefon IN VARCHAR2,
 vdireccio IN VARCHAR2,
 vcodipostal IN VARCHAR2,
 vciutat IN VARCHAR2,
 vlocalitat IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_ProfessorAlta'';
vParametres := vidinstitut || '','' || vnom || '','' || vcognom1 || '','' || vcognom2 || '','' || vdni;
vParametres := vParametres || '','' || vtelefon || '','' || vdireccio || '','' || vcodipostal || '','' || vciutat
|| '','' || vlocalitat;

IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;
IF (vcognom1 IS NULL OR vcognom1 = '''') THEN RAISE debe_rellenarse; END IF;
IF (vcognom2 IS NULL OR vcognom2 = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into PROFESSORS (idinstitut, nom, cognom1, cognom2, dni, telefon, direccio, codipostal, ciutat,
localitat)
VALUES (vidinstitut, vnom, vcognom1, vcognom2, vdni, vtelefon, vdireccio, vcodipostal, vciutat,
vlocalitat);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
```

```
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_ProfessorBaixa (vid IN VARCHAR2, RSP
OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_ProfessorBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM PROFESSORS WHERE idProfessor = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';

--Modificacio---------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_ProfessorModif
(
 vid IN NUMBER,
 vidinstitut IN NUMBER,
 vnom IN VARCHAR2,
 vcognom1 IN VARCHAR2,
 vcognom2 IN VARCHAR2,
 vdni IN VARCHAR2,
 vtelefon IN VARCHAR2,
 vdireccio IN VARCHAR2,
 vcodipostal IN VARCHAR2,
 vciutat IN VARCHAR2,
 vlocalitat IN VARCHAR2,
 RSP OUT Varchar2
)
```

```
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_ProfessorModif'';
vParametres :=  vid || '','' || vidinstitut || '','' || vnom || '','' || vcognom1 || '','' || vcognom2 || '','' ||
vdni;
vParametres := vParametres || '','' || vtelefon || '','' || vdireccio || '','' || vcodipostal || '','' || vciutat
|| '','' || vlocalitat;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE PROFESSORS SET
idinstitut = vidinstitut,
nom = vnom,
cognom1 = vcognom1,
cognom2 = vcognom2,
dni = vdni,
telefon = vtelefon,
direccio = vdireccio,
codipostal = vcodipostal,
ciutat = vciutat,
localitat = vlocalitat
WHERE idprofessor = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';


--CURSOS
--Alta----------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_CursAlta
(
 vidprofessor IN NUMBER,
 vidinstitut IN NUMBER,
 vnom IN VARCHAR2,
 vanyo IN NUMBER,
 RSP OUT Varchar2
```

```
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_CursAlta'';
vParametres := vidprofessor || '','' || vidinstitut || '','' || vnom || '','' || vanyo;

IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;
IF (vanyo IS NULL OR vanyo = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into CURSOS (idprofessor, idinstitut, nom, anyo)
VALUES (vidprofessor, vidinstitut, vnom, vanyo);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_CursBaixa (vid IN VARCHAR2, RSP OUT
Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_CursBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM CURSOS WHERE idCurs = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
```

```
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';

--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_CursModif
(
 vid IN NUMBER,
 vidprofessor IN NUMBER,
 vidinstitut IN NUMBER,
 vnom IN VARCHAR2,
 vanyo IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_CursModif'';
vParametres :=  vid || '','' || vidprofessor || '','' || vidinstitut || '','' || vnom || '','' || vanyo;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;
IF (vanyo IS NULL OR vanyo = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE CURSOS SET
idprofessor = vidprofessor,
idinstitut = vidinstitut,
nom = vnom,
anyo = vanyo
WHERE idcurs = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';
```

```
--ASSIGNATURES
--Alta---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssignaturaAlta
(
 vnom IN VARCHAR2,
 vdescripcio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AssignaturaAlta'';
vParametres := vnom || '','' || vdescripcio;

IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into ASSIGNATURES (nom, descripcio)
VALUES (vnom, vdescripcio);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssignaturaBaixa (vid IN VARCHAR2, RSP
OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_AssignaturaBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM ASSIGNATURES WHERE idAssignatura = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
```

```
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';

--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssignaturaModif
(
 vid IN NUMBER,
 vnom IN VARCHAR2,
 vdescripcio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AssignaturaModif'';
vParametres :=  vid || '','' || vnom || '','' || vdescripcio;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vnom IS NULL OR vnom = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE ASSIGNATURES SET
nom = vnom,
descripcio = vdescripcio
WHERE idassignatura = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
```

```
END;';


--ASSIGNATURESCURSOS
--Alta---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssigCursosAlta
(
 vIdCurs IN NUMBER,
 vIdAssignatura IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AssigCursosAlta'';
vParametres := vidcurs || '','' || vidassignatura;

IF (vIdCurs IS NULL OR vIdCurs = '''') THEN RAISE debe_rellenarse; END IF;
IF (vIdAssignatura IS NULL OR vIdAssignatura = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into ASSIGNATURESCURSOS (IdCurs, IdAssignatura) values (vIdCurs, vIdAssignatura);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssigCursosBaixa
(
 vIdCurs IN NUMBER,
 vIdAssignatura IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_AssigCursosBaixa'';
IF (vIdCurs IS NULL OR vIdCurs = '''') THEN RAISE debe_rellenarse; END IF;
```

```
IF (vIdAssignatura IS NULL OR vIdAssignatura = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM ASSIGNATURESCURSOS WHERE idCurs = vIdCurs and idAssignatura = vIdAssignatura;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vidCurs || '' '' || vidAssignatura, RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vidCurs || '' '' || vidAssignatura, RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vidCurs || '' '' || vidAssignatura, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vidCurs || '' '' || vidAssignatura, RSP);
END;';

--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssigCursosModif
(
 vIdCurs IN NUMBER,
 vIdAssignatura IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AssigCursosModif'';
vParametres := vidcurs || '','' || vidassignatura;

IF (vIdCurs IS NULL OR vIdCurs = '''') THEN RAISE debe_rellenarse; END IF;
IF (vIdAssignatura IS NULL OR vIdAssignatura = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE ASSIGNATURESCURSOS SET
idCurs = vIdCurs,
idAssignatura = vIdAssignatura
WHERE idCurs = vIdCurs and idAssignatura = vIdAssignatura;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
```

```
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';



--ASSIGNATURESPROFESSOR
--Alta----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssigProfAlta
(
 vIdProfessor IN NUMBER,
 vIdAssignatura IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AssigProfAlta'';
vParametres := vIdProfessor || '','' || vidassignatura;

IF (vIdProfessor IS NULL OR vIdProfessor = '''') THEN RAISE debe_rellenarse; END IF;
IF (vIdAssignatura IS NULL OR vIdAssignatura = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into ASSIGNATURESPROFESSORS (IdProfessor, IdAssignatura) values (vIdProfessor,
vIdAssignatura);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssigProfBaixa
(
 vIdProfessor IN NUMBER,
 vIdAssignatura IN NUMBER,
```

```
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_AssigProfBaixa'';
IF (vIdProfessor IS NULL OR vIdProfessor = '''') THEN RAISE debe_rellenarse; END IF;
IF (vIdAssignatura IS NULL OR vIdAssignatura = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM ASSIGNATURESPROFESSORS WHERE IdProfessor = vIdProfessor and idAssignatura =
vIdAssignatura;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdProfessor || '' '' || vidAssignatura, RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdProfessor || '' '' || vidAssignatura, RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdProfessor || '' '' || vidAssignatura, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdProfessor || '' '' || vidAssignatura, RSP);
END;';

--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_AssigProfModif
(
 vIdProfessor IN NUMBER,
 vIdAssignatura IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_AssigProfModif'';
vParametres := vIdProfessor || '','' || vidassignatura;

IF (vIdProfessor IS NULL OR vIdProfessor = '''') THEN RAISE debe_rellenarse; END IF;
IF (vIdAssignatura IS NULL OR vIdAssignatura = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE ASSIGNATURESPROFESSORS SET
IdProfessor = vIdProfessor,
idAssignatura = vIdAssignatura
WHERE IdProfessor = vIdProfessor and idAssignatura = vIdAssignatura;

RSP := ''OK'';
```

```
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';



--CALENDARI
--Alta---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_CalendariAlta
(
 vidcurs IN NUMBER,
 vidassignatura IN NUMBER,
 vdiasetmana IN NUMBER,
 vhorainici IN NUMBER,
 vhorafi IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
solapament EXCEPTION;
vValor VARCHAR(255 CHAR);
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_CalendariAlta'';
vParametres := vidcurs || '','' || vidassignatura || '','' || vdiasetmana || '','' || vhorainici || '','' ||
vhorafi;

IF (vdiasetmana IS NULL OR vdiasetmana = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhorainici IS NULL OR vhorainici = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhorafi IS NULL OR vhorafi = '''') THEN RAISE debe_rellenarse; END IF;

--Comprobem si existeixen solapacions
BEGIN
  SELECT idCalendari INTO vValor from Calendari Where idcurs = vidcurs and diasetmana = vdiasetmana
and
   ((horainici = vhorainici or horafi = vhorafi ) or  (horainici < vhorainici and horafi > vhorafi ) or
    (horainici > vhorainici and horainici < vhorafi ) or (horafi > vhorainici and horafi < vhorafi ));
  RAISE solapament;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
  Insert Into CALENDARI (idcurs, idassignatura, diasetmana, horainici, horafi)
```

```
 VALUES (vidcurs, vidassignatura, vdiasetmana, vhorainici, vhorafi);
 RSP := ''OK'';
 INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;

EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN solapament THEN
RSP := ''ERROR: La data a inserir provocaria un solapament de hores'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';


--Baixa---------------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_CalendariBaixa (vid IN VARCHAR2, RSP
OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_CalendariBaixa'';
IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM CALENDARI WHERE idCalendari = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vid,
RSP);
END;';
```

```
--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_CalendariModif
(
 vid IN NUMBER,
 vidcurs IN NUMBER,
 vidassignatura IN NUMBER,
 vdiasetmana IN NUMBER,
 vhorainici IN NUMBER,
 vhorafi IN NUMBER,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_CalendariModif'';
vParametres :=  vid || '','' || vidcurs || '','' || vidassignatura || '','' || vdiasetmana || '','' || vhorainici ||
'','' || vhorafi;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdiasetmana IS NULL OR vdiasetmana = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhorainici IS NULL OR vhorainici = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhorafi IS NULL OR vhorafi = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE CALENDARI SET
idcurs = vidcurs,
idassignatura = vidassignatura,
diasetmana = vdiasetmana,
horainici = vhorainici,
horafi = vhorafi
WHERE idcalendari = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';


--DIESFESTIUS
--Alta----------------------------------------------------------------
```

```
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_DiesFestiusAlta
(
 vdiaFestiu DATE,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_DiesFestiusAlta'';
vParametres := vdiaFestiu;

IF (vdiaFestiu IS NULL OR vdiaFestiu = '''') THEN RAISE debe_rellenarse; END IF;

Insert Into DIESFESTIUS (diaFestiu)
VALUES (vdiaFestiu);

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';


--Baixa-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_DiesFestiusBaixa (vIdDiaFestiu IN
NUMBER, RSP OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_DiesFestiusBaixa'';
IF (vIddiaFestiu IS NULL OR vIddiaFestiu = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM DIESFESTIUS WHERE IddiaFestiu= vIddiaFestiu;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdDiaFestiu, RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
```

```
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdDiaFestiu, RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdDiaFestiu, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vIdDiaFestiu, RSP);
END;';


--Modificacio----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_DiesFestiusModif
(
 vId IN NUMBER,
 vdiaFestiu IN DATE,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_DiesFestiusModif'';
vParametres :=  vid || '','' || vdiaFestiu;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdiaFestiu IS NULL OR vdiaFestiu = '''') THEN RAISE debe_rellenarse; END IF;

UPDATE DIESFESTIUS SET
diafestiu = vdiafestiu
WHERE iddiafestiu = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';
```

```
--HORESATENCIO
--Alta-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_HoresAtencioAlta
(
 vIdProfessor IN NUMBER,
 vdiasetmana IN NUMBER,
 vhoraInici IN NUMBER,
 vhoraFi IN NUMBER,
 vtipusHoraAtencio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
solapament EXCEPTION;
vValor VARCHAR(255 CHAR);
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_HoresAtencioAlta'';
vParametres :=  vIdProfessor || '','' || vdiasetmana || '','' || vhoraInici || '','' || vhoraFi || '','' ||
vtipusHoraAtencio;

IF (vIdProfessor IS NULL OR vIdProfessor = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdiasetmana IS NULL OR vdiasetmana = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhoraInici IS NULL OR vhoraInici = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhoraFi IS NULL OR vhoraFi = '''') THEN RAISE debe_rellenarse; END IF;
IF (vtipusHoraAtencio IS NULL OR vtipusHoraAtencio= '''') THEN RAISE debe_rellenarse; END IF;

--Comprobem si existeixen solapacions
BEGIN
  SELECT idhoraatencio INTO vValor from HORESATENCIO Where idprofessor = vidprofessor and
diasetmana = vdiasetmana and
   ((horainici = vhorainici or horafi = vhorafi ) or  (horainici < vhorainici and horafi > vhorafi ) or
    (horainici > vhorainici and horainici < vhorafi ) or (horafi > vhorainici and horafi < vhorafi ));
  RAISE solapament;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
  Insert Into HORESATENCIO (IdProfessor, diasetmana, horaInici, horaFi, tipusHoraAtencio)
  VALUES (vIdProfessor, vdiasetmana, vhoraInici, vhoraFi, vtipusHoraAtencio);
  RSP := ''OK'';
  INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;

EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut insertar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN solapament THEN
RSP := ''ERROR: La data a inserir provocaria un solapament de hores'';
```

```
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar d´alta'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

--Baixa-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_HoresAtencioBaixa (vId IN NUMBER, RSP
OUT Varchar2)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
BEGIN
vProcedimiento := ''proc_HoresAtencioBaixa'';
IF (vId IS NULL OR vId = '''') THEN RAISE debe_rellenarse; END IF;

DELETE FROM HORESATENCIO WHERE IdHoraAtencio = vId;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vId,
RSP);
EXCEPTION
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vId,
RSP);
WHEN NO_DATA_FOUND THEN
RSP := ''ERROR: No s´ha trobat el registre'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vId,
RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al donar de baixa'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE, vId,
RSP);
END;';

--Modificacio-----------------------------------------------------------------
EXECUTE IMMEDIATE 'CREATE OR REPLACE PROCEDURE proc_HoresAtencioModif
(
 vId IN NUMBER,
 vIdProfessor IN NUMBER,
 vdiasetmana IN NUMBER,
 vhoraInici IN NUMBER,
 vhoraFi IN NUMBER,
 vtipusHoraAtencio IN VARCHAR2,
 RSP OUT Varchar2
)
IS
debe_rellenarse EXCEPTION;
vProcedimiento VARCHAR2(25);
vParametres VARCHAR2(255);
BEGIN
vProcedimiento := ''proc_HoresAtencioModif'';
```

```
vParametres :=  vid || '','' || vIdProfessor || '','' || vdiasetmana || '','' || vhoraInici || '','' || vhoraFi ||
'','' || vtipusHoraAtencio;

IF (vid IS NULL OR vid = '''') THEN RAISE debe_rellenarse; END IF;
IF (vIdProfessor IS NULL OR vIdProfessor = '''') THEN RAISE debe_rellenarse; END IF;
IF (vdiasetmana IS NULL OR vdiasetmana = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhoraInici IS NULL OR vhoraInici = '''') THEN RAISE debe_rellenarse; END IF;
IF (vhoraFi IS NULL OR vhoraFi = '''') THEN RAISE debe_rellenarse; END IF;
IF (vtipusHoraAtencio IS NULL OR vtipusHoraAtencio= '''') THEN RAISE debe_rellenarse; END IF;

UPDATE HORESATENCIO SET
IdProfessor = vIdProfessor,
diasetmana = vdiasetmana,
horaInici = vhoraInici,
horaFi = vhoraFi,
tipusHoraAtencio = vtipusHoraAtencio
WHERE IdHoraAtencio = vid;

RSP := ''OK'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida)
VALUES (vProcedimiento, SYSDATE, vParametres, RSP);
EXCEPTION
WHEN STORAGE_ERROR THEN
RSP := ''ERROR: No s´ha pogut modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN debe_rellenarse THEN
RSP := ''ERROR: No s´han emplenat els paràmetres necessaris'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
WHEN OTHERS THEN
RSP := ''ERROR: Error genéric al modificar'';
INSERT INTO LOGS (nom, data, paramEntrada, paramSortida) VALUES (vProcedimiento, SYSDATE,
vParametres, RSP);
END;';

END;
/
```