

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

Memòria



Autor: Daniel González Chorro

Consultor: Ismael Pérez Laguna

Resum

Aquest projecte és una proposta de disseny i implementació d'una base de dades relacional per la Generalitat de Catalunya. Aquest client desitja implantar una aplicació de gestió d'amonestacions i sancions en els centres educatius catalans d'ensenyament secundari.

Per triar el proveïdor tecnològic que s'encarregarà d'implementar el sistema, la Generalitat ha convocat un concurs públic on els interessats podran presentar el seu projecte.

El treball que a continuació s'exposa compleix completament amb tots els requeriments sol·licitats a les bases del concurs públic.

Aquest document és la memòria, i tan sols és un dels tres que s'entregaran al client. Els altres dos són el producte, constituït per una sèrie de scripts en llenguatge PL/SQL d'Oracle, i la presentació que sintetitza el treball elaborat.

Al començament de la memòria trobarem el capítol d'introducció, el qual es centra bàsicament en la realització d'un pla de treball realista que permeti dur a terme el projecte amb èxit.

Al segon capítol es presenta la fase de disseny que tot sistema de programari incorpora. En el context d'aquest projecte, la fase de disseny consisteix en realitzar els esquemes conceptual, lògic i físic de la base de dades.

Una vegada s'ha finalitzat l'etapa de disseny, es procedeix amb la fase d'implementació. En aquesta fase es desenvoluparan les solucions que donaran resposta a totes les necessitats del client.

Al quart capítol s'exposa els recursos econòmics, materials, tècnics i humans necessaris per elaborar el projecte.

Finalment la memòria terminarà amb les conclusions de l'autor un cop s'ha dut a terme tot el treball i amb la inclusió del glossari i bibliografia emprats.

Àrea del TFC: Bases de dades relacionals

Paraules clau: BD (Base de Dades), Model Entitat-Interrelació (E/R), Model Relacional, PL/SQL (Procedural Language/Structured Query Language), script, SGBD (Sistema de Gestió de Base de Dades), SQL (Structured Query Language), Stored Procedure (Procediment Emmagatzemat), Trigger (Disparador) i UML (Unified Modeling Language).

Índex de continguts

1. Introducció.....	6
1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.....	6
1.2 Objectius del TFC.....	7
1.3 Enfocament i mètode seguit.....	8
1.4 Planificació del projecte.....	9
1.4.1 Dates clau.....	9
1.4.2 Valoració temporal i disponibilitat.....	9
1.4.3 Planificació proposada i diagrama de Gantt	10
1.4.4 Estimació en els lliuraments.....	11
1.4.5 Anàlisi de riscos i mesures de prevenció	12
1.5 Productes obtinguts.....	13
1.6 Descripció dels altres capítols de la memòria.....	14
2. Disseny de la Base de Dades.....	15
2.1 Disseny conceptual: Model E/R	15
2.1.1 Justificació i restriccions de l'esquema E/R	17
2.2 Disseny lògic: Model relacional.....	21
2.2.1 Justificació i restriccions de l'esquema relacional.....	23
2.3 Disseny físic	37
3. Implementació de la Base de Dades.....	39
3.1 Aspectes generals dels scripts.....	40
3.1.1 Capçalera descriptiva.....	40
3.1.2 Tractament de les dades d'entrada.....	41
3.1.3 Tractament d'excepcions	41
3.1.4 Emmagatzemament dels registres de log.....	43

3.2 Aspectes específics dels scripts.....	44
3.2.1 Script de creació de taules	44
3.2.2 Script de càrrega de dades	44
3.2.3 Procediments d'ABM d'alumnes i professors.....	45
3.2.4 Procediments d'ABM dels cursos, assignatures i calendari escolar.....	46
3.2.5 Procediments d'ABM de les tipologies d'amonestacions i sancions.....	47
3.2.6 Procediments d'ABM d'amonestacions.....	48
3.2.7 Procediments d'ABM de sancions	49
3.2.8 Gestió del Mòdul Estadístic.....	50
3.2.9 Procediments de consulta	51
3.3 Joc de proves i errors.....	52
4. Recursos necessaris i valoració econòmica	53
4.1 Recursos necessaris	53
4.1.1 Recursos materials.....	53
4.1.2 Recursos tècnics	54
4.1.3 Recursos humans	55
4.2 Valoració econòmica	56
5. Conclusions	57
6. Glossari.....	58
7. Bibliografia	59

Índex de figures

Figura 1- Diagrama de Gantt.....	10
Figura 2- Model E/R	16
Figura 3- Model E/R Segment 1	17
Figura 4- Model E/R Segment 2.....	17
Figura 5- Model E/R Segment 3.....	18
Figura 6- Model E/R Segment 4.....	18
Figura 7- Model E/R Segment 5.....	19
Figura 8- Model E/R Segment 6.....	19
Figura 9- Model E/R Segment 7.....	20
Figura 10- Model Relacional Part 1	22
Figura 11- Model Relacional Part 2.....	23
Figura 12- Índexs per l'optimització.....	38
Figura 13- Capçalera descriptiva	40
Figura 14- Tractament de les dades d'entrada	41
Figura 15- Detecció d'excepcions	42
Figura 16- Tractament d'excepcions	42
Figura 17- Registre de log	43
Figura 18- Aspecte taula de log	43
Figura 19- Aspecte taula TipusAmonestacio	47
Figura 20- Aspecte taula TipusSancio	47
Figura 21- Crides als procediments d'alta del mòdul estadístic.....	50
Figura 22- Crides als procediments de baixa del mòdul estadístic.....	51
Figura 23- Exemple de resultat retornat per una consulta.....	52

1. Introducció

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

La Generalitat de Catalunya desitja implantar una aplicació de gestió d'amonestacions i sancions en els centres educatius catalans d'ensenyament secundari. Aquesta aplicació requereix com a base del seu funcionament una Base de Dades (BD) que serveixi com a magatzem d'informació.

La Generalitat ha convocat un concurs públic on els proveïdors tecnològics interessats podran presentar el seu projecte. Els participants només hauran de presentar la proposta de la BD, donat que l'aplicació serà definida posteriorment en una segona fase, la qual no forma part del nostre objecte d'estudi.

Així doncs, el nucli central del nostre projecte serà el disseny i implementació d'una base de dades relacional, que haurà de complir amb una sèrie de requeriments descrits pel client. La BD guardarà la informació relativa a les amonestacions i sancions de tots els alumnes que estudien als centres d'ensenyament secundari de Catalunya.

Per la temàtica del nostre objecte d'estudi, la base de dades emmagatzemarà informació relativa als centres educatius, als alumnes, professorat, assignatures, amonestacions i sancions, entre d'altra tipus d'informació associada a aquest àmbit.

La informació que contindrà la BD serà utilitzada per cadascun dels diversos instituts d'ensenyament catalans. És a dir, el disseny sol·licitat haurà de permetre que un número indeterminat d'instituts pugui manipular (inserir, eliminar, modificar) i consultar simultàniament la informació emmagatzemada al sistema.

Perquè la nostra proposta tingui les majors opcions possibles de ser triada com a guanyadora, haurà de complir amb tots els requeriments especificats per part del client (Generalitat). A més, haurà de ser tan econòmica com sigui possible. És valorarà també la qualitat i l'eficiència del disseny proposat. Finalment la realització d'una bona presentació del producte serà un altre dels factors clau per assolir l'èxit.

La Generalitat ha establert unes bases en el concurs que tota proposta haurà de complir rigorosament. Aquestes bases queden dividides per dues categories clarament diferenciades: Els requeriments funcionals i els de metodologia.

En els primers, el client ha descrit punt a punt cadascuna de les funcionalitats que ha de tenir l'aplicació. La nostra tasca serà donar una solució òptima a cadascun dels requeriments. A més, sempre que sigui possible intentarem innovar en la nostra solució. No oblidem que un dels objectius és destacar positivament per sobre de la competència.

Respecte els requeriments de metodologia, també s'han marcat unes pautes que hauran de complir cadascuna de les propostes. Un resum de les més rellevants és:

- Disseny de la BD mitjançant un diagrama Entitat/Relació.
- Creació dels scripts de creació de taules, índexs, disparadors, etc.
- Implementació dels procediments necessaris per realitzar les funcionalitats descrites.
- Documentació del codi i realització d'un manual d'ús d'usuari.
- Utilització del sistema gestor de bases de dades Oracle.
- Creació d'un joc de proves que serveixi per fer el testeig del sistema i la demostració del seu funcionament.
- Tractament exhaustiu dels errors que puguin sorgir en la interacció amb el sistema.

1.2 Objectius del TFC

L'objectiu principal d'aquest Treball Final de Carrera és consolidar i ampliar els coneixements adquirits al llarg dels estudis d'Enginyeria Tècnica en Informàtica en l'àrea de bases de dades. Concretament, haurem de sintetitzar els coneixements adquirits en les assignatures de Bases de dades I, Bases de dades II, Sistemes Gestors de Bases de Dades i Enginyeria del Programari per realitzar un projecte vinculat a l'exercici professional de la informàtica.

Per dur a terme aquest objectiu, haurem de presentar el disseny i implementació d'una base de dades per la Conselleria d'Ensenyament del nou govern de la Generalitat de Catalunya. Sota el context de que som un proveïdor de solucions informàtiques especialitzat en l'àrea de les bases de dades, haurem d'elaborar la millor proposta possible amb l'objectiu de guanyar el concurs públic obert per la Generalitat.

A banda de l'objectiu principal que hem descrit, hi ha una sèrie d'objectius més específics que també es pretenen assolir amb l'elaboració del projecte. Aquests són:

- **Aprendre a gestionar un projecte:** Saber com començar a abordar-ho, ser conscient de la importància que té l'elaboració d'un bon pla de treball, saber afrontar les dificultats que puguin sorgir al llarg del projecte, aprendre a fer un informe formal de manera correcta i sintetitzar els coneixements en una presentació virtual.
- **Ampliar els coneixements d'Oracle:** Tenir l'oportunitat de millorar i ampliar els coneixements del SGBD Oracle que es varen adquirir en l'assignatura de Sistemes Gestors de Bases de Dades. A nivell professional, Oracle avui en dia és un dels SGBD més estès del món.
- **Adquirir experiència en el camp de les bases de dades:** Aquest projecte ens permetrà dissenyar i implementar una BD pràcticament des de zero. A diferència d'altres assignatures de bases de dades on els coneixements s'adquiren mitjançant pràctiques i problemes molt específics, el TFC ens permetrà abordar un projecte de base de dades des del seu origen fins a la seva presentació.

1.3 Enfocament i mètode seguit

Com hem comentat amb anterioritat, el propòsit final d'aquest projecte és desenvolupar un programari donada una metodologia i uns requeriments. Com tot sistema de programari, aquest haurà de ser sotmès a una sèrie d'etapes successives que permetin garantir els terminis establerts i la qualitat del producte final, respectant els límits dels recursos assignats.

Aquesta metodologia de dividir les tasques en fases en l'àmbit del desenvolupament del programari, és el que es coneix com el *cicle de vida del programari*. Aquest cicle engloba totes les etapes, des de la fase inicial fins que el producte es completament finalitzat.

Hi ha diversos models del cicle de vida del programari, però si bé no hi ha cap que sigui ideal o universal. El model que seguirà aquest projecte és el conegut com a cicle de vida del *Rational Unified Process*, donat que actualment és la metodologia més estesa i que millor s'adapta a les característiques del programari a desenvolupar.

El model R.U.P. està constituït per les següents fases:

1. **Inici:** Es delimita l'abast del projecte, es realitza l'anàlisi de riscos i es justifiquen els recursos econòmics i materials.
2. **Elaboració:** Es fa l'estudi exhaustiu del problema tenint en compte les necessitats d'informació, els eventuais requisits no funcionals i les restriccions. S'estableix l'arquitectura general del programari i s'elabora una planificació del projecte.
3. **Construcció:** Es desenvolupa el producte de manera incremental (es procedeix per parts) i iterativa (es repeteixen els procediments per cada part). Té en compte tots el requeriments funcionals.
4. **Transició:** Lliurament del producte al client, començament de la seva utilització i posterior manteniment. (correcció d'errors, agregar noves funcionalitats, suport, etc)

A cadascuna de les fases descrites es poden dur a terme de manera iterativa els components següents de procés: Recollida de requisits, anàlisi i disseny, implementació i prova.

1.4 Planificació del projecte

Una bona divisió de la feina total associada al projecte en tasques i la correcta distribució d'aquestes al llarg del temps, és fonamental per garantir l'èxit de tot projecte.

El calendari de tasques sempre s'haurà d'adaptar a les dates d'entrega imposades pel client.

1.4.1 Dates clau

En el nostre cas haurem de complir de manera rigorosa la temporització definida pels consultors:

ACTIVITAT	DATA INICI	DATA ENTREGA
Inici projecte (PAC 1)	03/03/2011	20/03/2011
Control de la planificació 1 (PAC 2)	21/03/2011	17/04/2011
Control de la planificació 2 (PAC 3)	18/04/2011	15/05/2011
Fi projecte (LLIURAMENT FINAL)	16/05/2011	12/06/2011
Defensa del treball	20/06/2011	26/06/2011

1.4.2 Valoració temporal i disponibilitat

El nombre total de dies des del començament del projecte fins al seu lliurament és de 102 dies. (aproximadament unes 15 setmanes)

Degut a la meva situació personal, on tinc que treballar 40 hores setmanals i a més estic cursant les assignatures de *Seguretat en Xarxes de Computadors* i *Teoria d'autòmats i Llenguatges Formals II*, podré destinar entre 16 i 20 hores setmanals per l'elaboració d'aquest projecte.

Multiplicant la mitjana d'aquestes hores pel nombre total de setmanes que durarà el projecte, obtenim una estimació de 270 hores. A priori, aquest serà el temps total en hores necessari per dur a terme tot el treball.

1.4.3 Planificació proposada i diagrama de Gantt

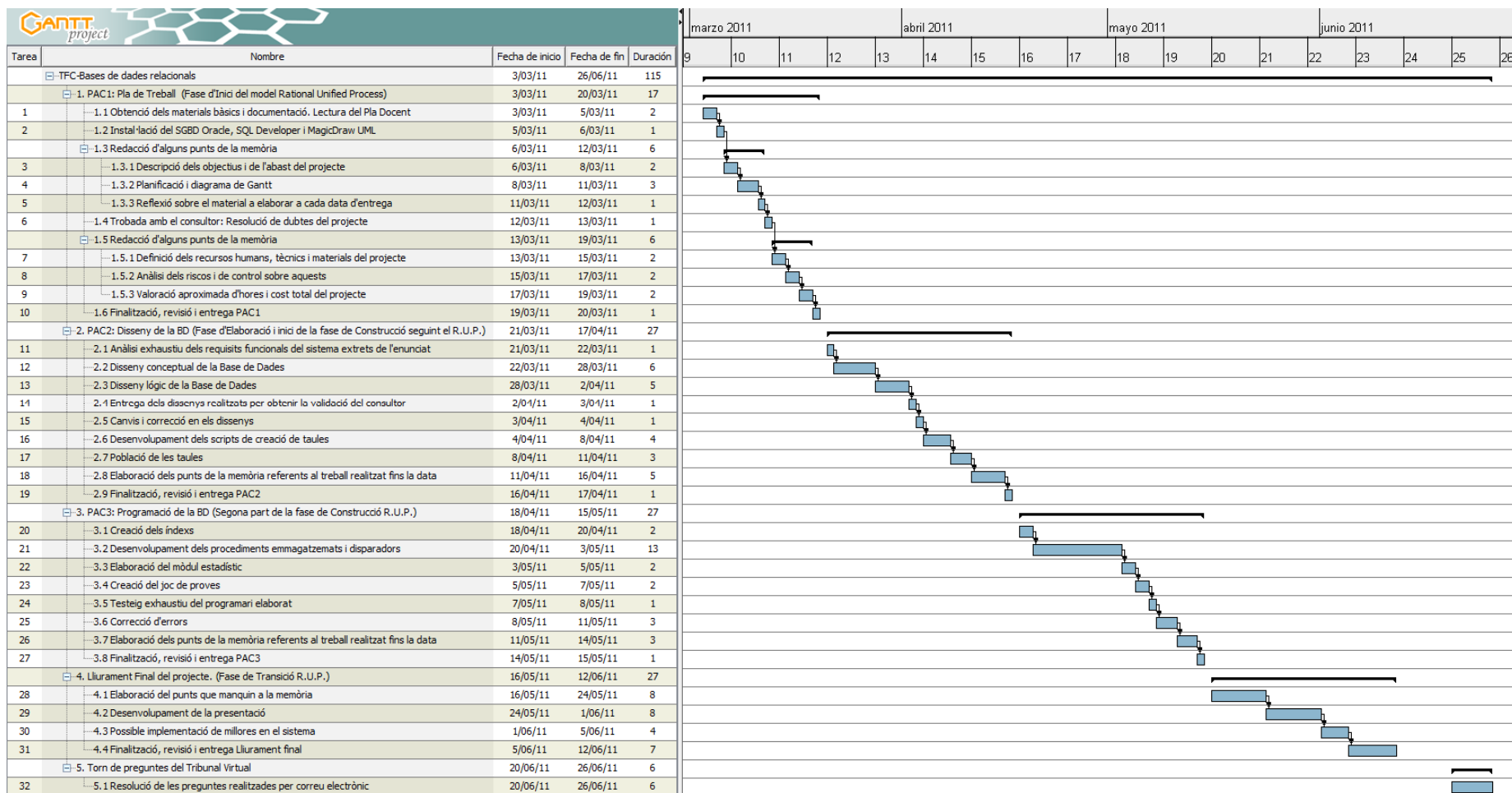


Figura 1

A la pàgina anterior, podem observar que a la part esquerra del gràfic es presenta la planificació del projecte mitjançant un calendari. Aquest marcarà el repartiment de les tasques associades a la realització del projecte al llarg del semestre.

A la part dreta del gràfic es pot observar el diagrama de Gantt associat a la planificació. Cada rectangle del diagrama correspon a una tasca independent, i aquestes s'han de dur a terme de manera seqüencial. Les línies negres representen tasques que requereixen de varies fases o subtasques per ser completades.

Hi ha algunes tasques que no queden recollides en la proposta del pla de treball, donat que no es pot predir quan es faran, com per exemple les cites bibliogràfiques. Aquestes al igual que la creació de l'annex, es realitzarà a mesura que sorgeixi la necessitat.

Finalment comentar que al hora de planificar el calendari de tasques s'ha fet sota una perspectiva molt negativa en referència a l'estimació de temps. És a dir, que hi ha tasques que haurien d'estar acabades uns dies abans de la previsió, sempre i quan no sorgeixi cap imprevist.

1.4.4 Estimació en els lliuraments

Al TFC de Bases de dades relacionals es tenen que realitzar quatre lliuraments. Segons la planificació anterior, a continuació definirem els continguts que s'entregaran a cadascun:

- **PAC1:** Es lliurarà el pla de treball. Aquest document esta constituït principalment per la descripció del projecte, la seva planificació, els recursos necessaris, l'anàlisi de riscos i la valoració econòmica.
- **PAC2:** Es lliurarà els dissenys conceptual i lògic de la BD. A més els scripts SQL de creació de taules i la seva població amb un joc de dades significatiu.
- **PAC3:** Es lliurarà els scripts que contenen els procediments emmagatzemats, disparadors i el joc de prova associat. També la part de la memòria corresponent al treball realitzat.
- **PAC4:** S'entregarà la memòria en el seu estat definitiu, una presentació que sintetitzi el treball realitzat, i el codi font final de la base de dades.

1.4.5 Anàlisi de riscos i mesures de prevenció

En aquest apartat tractarem els principals problemes o fets que podrien produir-se al llarg de la realització del projecte i que posarien en perill l'assoliment dels objectius o el compliment de la planificació inicial. També s'exposarà la mesura de prevenció que s'ha determinat per anul·lar o mitigar el problema en el cas que s'arribi a produir.

Llistarem els possibles riscos i mesures associades:

- **Risc de pèrdua de dades:** Podria passar que l'equip on s'està desenvolupant el projecte deixi de funcionar degut a una anomalia del subministrament elèctric o simplement si algun component hardware es fa malbé.

Mesura adoptada: Després de cada jornada diària de treball, es farà una còpia de seguretat de tota la informació del projecte i s'emmagatzemarà en un suport extern independent. (disc dur extern o memòria USB)

- **Risc de malaltia o imprevist:** Podria succeir que l'estudiant es posi malalt durant un nombre considerable de dies o li sorgeixi algun imprevist laboral o familiar i tingui que destinar menys dies dels previstos al projecte.

Mesura adoptada: La planificació del projecte s'ha realitzat tenint en compte aquets factors. El temps per la realització de cadascuna de les tasques s'ha determinat amb una perspectiva molt negativa. Hi ha tasques que en comptes d'una jornada sencera de treball hauria de fer-se en poques hores. El mateix succeeix amb les tasques que requereixen varis dies de treball, sinó sorgeixen imprevistos hauria de sobrar un cert temps.

- **Risc de no complir amb la planificació:** Podria succeir que l'estudiant no compleixi al llarg del semestre amb la planificació de tasques proposada i s'endarrereixi.

Mesura adoptada: Conscienciar-se de que el treball continu és la millor manera d'assolir l'èxit.

1.5 Productes obtinguts

Després de la realització d'aquest projecte s'ha obtingut un producte compost per una sèrie d'arxius. Aquests fitxers bàsicament són scripts PL/SQL que implementen totes les funcionalitats sol·licitades i que poden ser carregats directament al SGBD Oracle.

A continuació s'esmenten aquests fitxers i es descriu breument la seva finalitat donat que en altres capítols d'aquest document s'aprofundeix en el seu funcionament i a més els mateixos fitxers ja es troben documentats internament.

Llistat de fitxers que implementen la BD del sistema de gestió d'amonestacions i sancions en centres educatius:

Creacio de taules.sql : Script per crear totes les taules, triggers i seqüències vinculades a les claus artificials emprades per aquestes taules.

ABM alumne.sql : Script per crear els procediments d'ABM dels alumnes.

ABM amonestacions.sql : Script per crear els procediments d'ABM de les amonestacions.

ABM assignatura.sql : Script per crear els procediments d'ABM de les assignatures.

ABM calendari escolar.sql : Script per crear els procediments d'ABM del calendari escolar.

ABM curs.sql : Script per crear els procediments d'ABM dels cursos.

ABM professor.sql : Script per crear els procediments d'ABM dels professors.

ABM sancions.sql : Script per crear els procediments d'ABM de les sancions.

ABM tipus amonestacio.sql : Script per crear els procediments d'ABM de les tipologies d'amonestacions.

ABM tipus sancio.sql : Script per crear els procediments d'ABM de les tipologies de sancions.

Gestio Modul Estadistic.sql : Script per crear els procediments de gestió del Mòdul Estadístic.

Procediments de consulta.sql : Script per crear els procediments que donen resposta a les quatre consultes sol·licitades.

Triggers Gestio Sancions Automatiques.sql : Script format per quatre disparadors que implementen l'aplicació de sancions automàtiques.

Carrega de dades.sql : Script per omplir totes les taules amb un joc de proves de mida considerable.

Joc de proves i errors.sql : Sentències independents PL/SQL per realitzar proves i crear situacions d'excepció.

Adicionalment s'inclou el fitxer de text **INSTRUCCIONS.txt** el qual té com a finalitat donar les passes necessàries per executar i provar els scripts que s'acaben d'especificar.

D'altra banda, apart dels fitxers que conformen el producte també s'ha elaborat aquest document de text (**memòria.doc**). La memòria sintetitza els aspectes més rellevants del projecte, com són el pla de treball, la metodologia emprada per dur-lo a terme, el disseny implementat i la solució als requeriments plantejats.

Finalment, el treball es presenta mitjançant un document en format PowerPoint compost de diverses diapositives que sintetitzen el treball realitzat (**presentacio.ppt**).

1.6 Descripció dels altres capítols de la memòria

A més d'aquest primer capítol d'introducció, aquest document està format pels següents capítols:

Capítol 2- Disseny de la Base de Dades: Es tractaran els aspectes relacionats amb el disseny de la BD que emmagatzemarà tota la informació requerida pel sistema de gestió d'amonestacions i sancions. A part d'exposar el disseny triat, també es justificarà mitjançant la segmentació del disseny en diferents fragments de forma molt visual.

Capítol 3- Implementació de la Base de Dades: Recull els aspectes relacionats amb la implementació i programació de cadascuna de les funcionalitats. Es justificarà les diferents solucions aportades que permeten complir amb tots els requeriments sol·licitats pel client.

Capítol 4- Recursos necessaris i valoració econòmica: S'enumera i es descriuen cadascun dels recursos necessaris per poder dur a terme el projecte. També es realitza una estimació del cost real que suposaria portar el projecte a la realitat.

Capítol 5- Conclusions: Es valora el treball un cop s'ha finalitzat.

Finalment hi ha un glossari on es defineixen els termes més rellevants vinculats al projecte i una bibliografia que ha servit com a ajuda per culminar-ho.

2. Disseny de la Base de Dades

L'elaboració d'un bon disseny de la base de dades és fonamental quan es desitja que el sistema o programari que la vagi a utilitzar sigui eficient, robust, fàcilment actualitzable i en general que sigui de qualitat.

El disseny de la BD no és una tasca trivial, ben bé tot el contrari. S'ha de tenir en compte una infinitat de factors, com per exemple poden ser; la velocitat d'accés a la informació emmagatzemada, la mida de les dades, la facilitat per extreure la informació requerida, que satisfaci tots els requeriments d'informació dels usuaris, que es pugui modificar o ampliar amb facilitat, etc.

Amb l'objectiu d'obtenir una bona estructura de base de dades, en aquest treball s'ha optat per l'ús del mètode de disseny més universal, el qual consisteix en dividir el disseny de la BD en tres etapes clarament diferenciades. La primera etapa es coneguda com *l'etapa del disseny conceptual*, la segona és *l'etapa del disseny lògic* i finalment la tercera té el nom d'*etapa del disseny físic*.

A continuació es descriurà amb detall en que consisteix cadascuna de les etapes i es relacionaran amb el disseny del sistema de gestió d'amonestacions i sancions en centres educatius.

2.1 Disseny conceptual: Model E/R

El disseny conceptual principalment consisteix en l'elaboració d'una estructura d'informació (en aquest cas un esquema) independent de la tecnologia que s'emprarà posteriorment en la fase d'implementació de la base de dades. És a dir, aquesta fase no té en compte quin tipus de base de dades s'emprarà (relacional, jeràrquic, etc) ni tampoc amb quin Sistema Gestor de Base de Dades es treballarà.

El resultat d'aquesta primera etapa dona lloc a una estructura d'informació que ha de ser expressada mitjançant un model. El model que s'utilitzarà és el conegut com a *model entitat-interrelació* (entity-relationship o E/R) donat que és molt simple, fàcil d'entendre i el més estès. El model E/R està format fonamentalment per les entitats, els atributs i les interrelacions.

En aquest treball s'ha optat per emprar la notació UML (Unified Modeling Language) per a l'elaboració del diagrama E/R.

A continuació es mostrarà l'esquema conceptual associat al sistema de gestió d'amonestacions i sancions en centres educatius i posteriorment es justificarà el disseny exposat mitjançant la segmentació d'aquest en parts amb les seves corresponents explicacions.

Model E/R

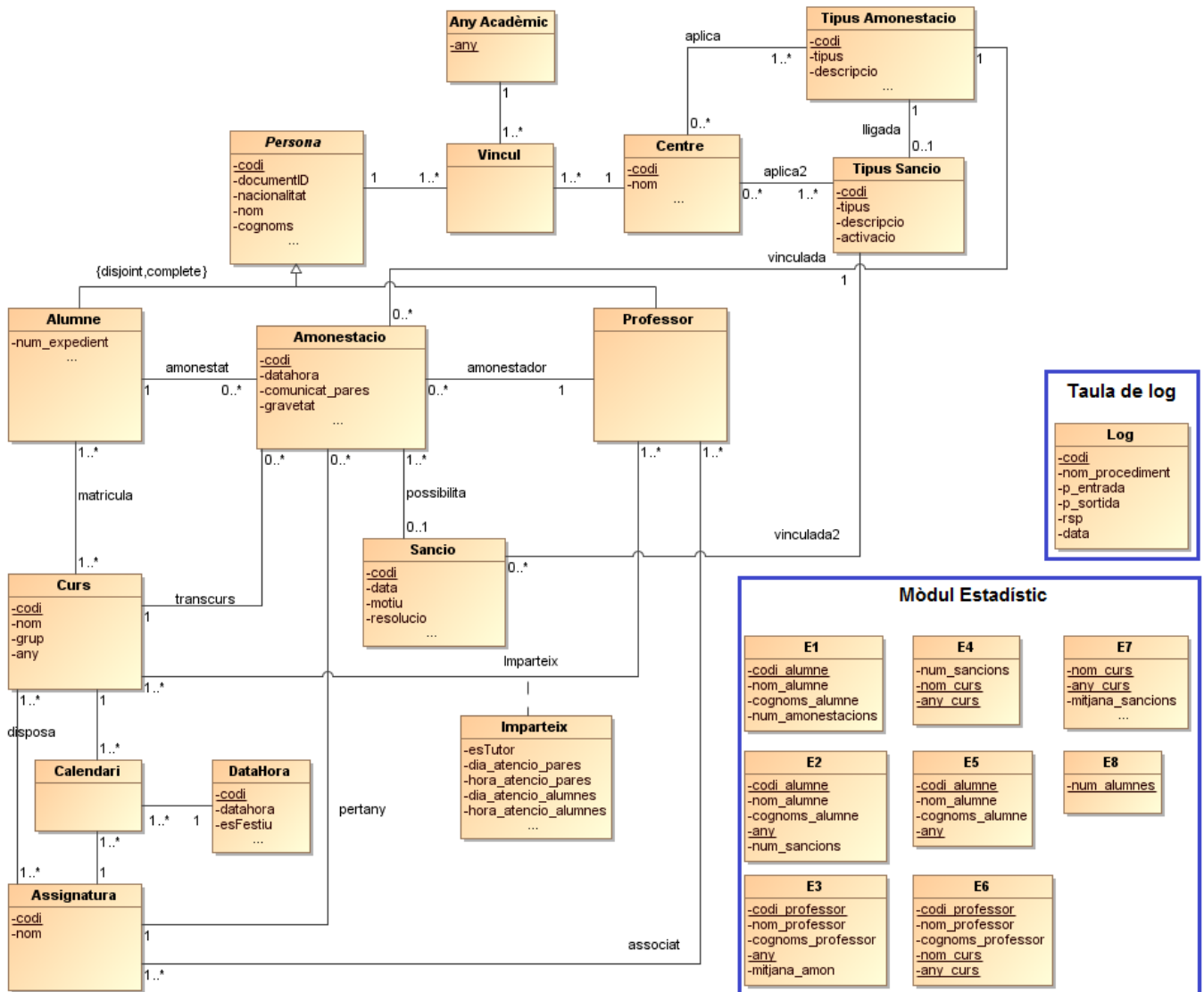


Figura 2

Informació a tenir en compte sobre l'esquema:

- Les claus primàries s'indiquen mitjançant el subratllat de l'atribut.
- L'esquema no contempla les claus alternatives. L'esquema lògic si que les especifica.
- S'ha emmarcat la taula log i les taules del mòdul estadístic per expressar que són independents.
- A l'esquema se li ha aplicat una reificació (transformació de les classes associatives) de les associacions n-àries amb grau major que dos. (n>2)

2.1.1 Justificació i restriccions de l'esquema E/R

En aquest apartat es donarà una justificació a les decisions preses alhora de crear el disseny conceptual i s'exposarà les restriccions associades a aquest. Per simplicitat, es segmentarà l'esquema E/R anteriorment mostrat i es procedirà a donar una explicació de cada segment. (s'entén per segment una part específica de l'esquema conceptual)

Segment 1:

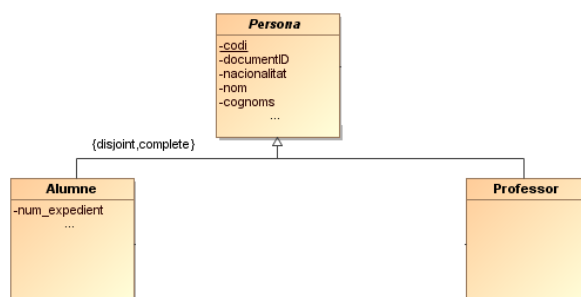


Figura 3

Aquesta part del diagrama posa de manifest la idea de que l'entitat *Professor* i l'entitat *Alumne* són subclasses de la superclasse *Persona*. Aquesta generalització és de tipus *disjunta/total*, el que significa que una mateixa persona no pot ser alhora estudiant i professor. A més qualsevol persona ha de ser o bé alumne o bé professor, no es contempla un altre tipus.

S'ha definit la classe *Persona* com a tipus *abstract*, amb l'objectiu d'explicitar la idea de que no es poden crear ocurrences directament a la classe *Persona*, obligatòriament es tenen que crear en una de les seves subclasses.

Segment 2:

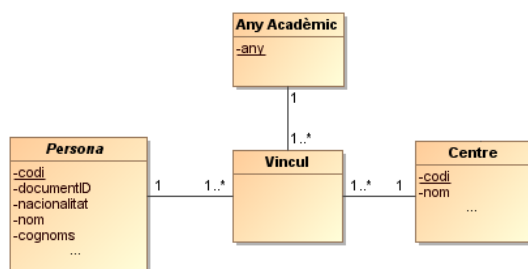
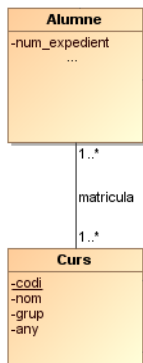


Figura 4

Aquest fragment de diagrama recull la idea de que una persona (ja sigui estudiant o professor) pot tenir una o més vinculacions al llarg dels anys, i a més pot canviar de centre.

A un centre sempre ha d'estar vinculada una persona com a mínim, tot i que lògicament hauria de haver-hi moltes més. Per últim, una vinculació sempre estarà formada per una persona, que es troba en un centre a un determinat any.

Segment 3:

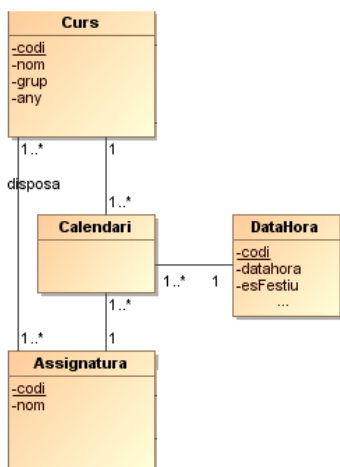


Un alumne sempre s'haurà de matricular com a mínim a un curs, tot i que al llarg del temps es pot matricular en diferents cursos.

Un curs sempre estarà format com a mínim per un alumne, però el cas més freqüent és que en un mateix curs es matriculin molts estudiants.

Figura 5

Segment 4:



Un curs està format per una o varies assignatures. Una assignatura com a mínim s'impartirà a un curs, tot i que el més freqüent és que s'imparteixi en varis.

D'altra banda cada curs disposa d'un calendari. Aquest especifica que una assignatura s'imparteix a un determinat curs, en un dia i a una hora en concret. El cas més freqüent és que la taula calendari disposi de moltes ocurrències per un mateix curs.

Figura 6

Segment 5:

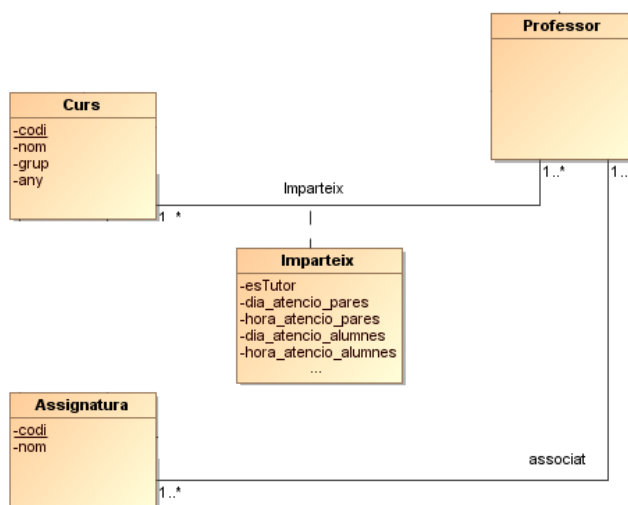


Figura 7

Aquest segment de diagrama està compost per dues interrelacions independents. La interrelació *Imparteix* expressa el significat de que un professor pot impartir classe a un curs o a més. D'altra banda també defineix que un determinat curs sempre disposarà d'un professor com a mínim.

La interrelació *Associat* s'empra per modelar el fet de que un professor sempre impartirà una assignatura com a mínim, i conseqüentment una assignatura sempre serà impartida per un o més professors.

Segment 6:

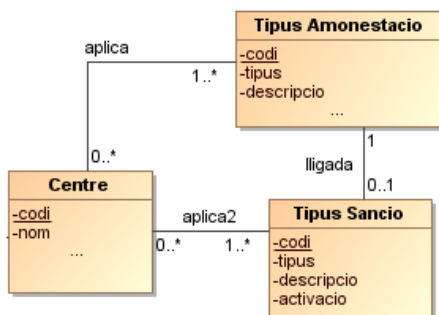


Figura 8

Les interrelacions *Aplica* i *Aplica2* s'utilitzen per modelitzar que un centre educatiu pot definir un règim d'amonestacions i de sancions en particular, respectivament.

La interrelació *Lligada* s'empra per indicar que cada sanció ha d'anar lligada a un tipus d'amonestació definit.

Segment 7:

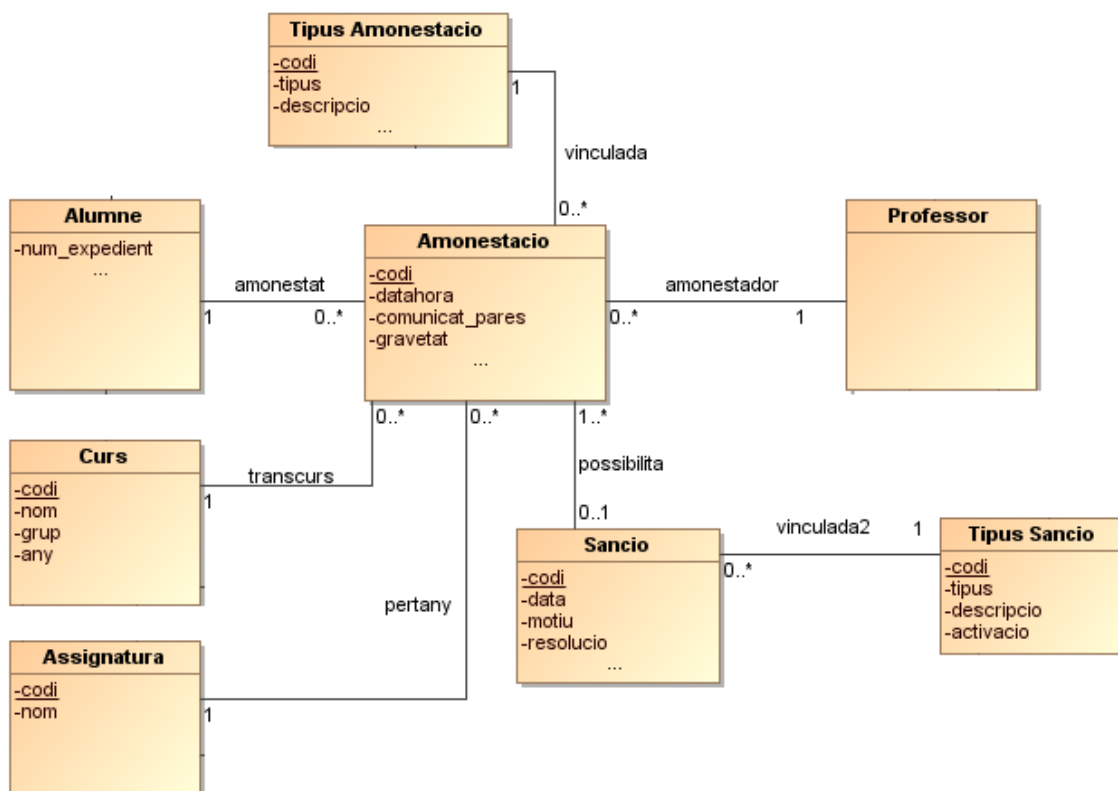


Figura 9

En l'últim fragment de diagrama s'observen moltes entitats e interrelacions. Tot i que sembla una mica enrevessat, la idea que modelitza aquest segment és simple.

Una amonestació sempre tindrà a un alumne i a un professor com a protagonistes. A més aquesta succeirà en un curs i en una assignatura determinada, i serà d'un dels tipus predefinits prèviament.

D'altra banda es pot observar com una amonestació possibilita la creació d'una sanció (0..1). Tota sanció anirà lligada a una amonestació greu o una acumulació d'amonestacions. Finalment, el diagrama també especifica que cada sanció serà d'un dels tipus predefinits prèviament.

Per concloure aquest apartat de disseny conceptual, indicar que la justificació del segment que correspon a les taules del mòdul estadístic i a la taula log, serà exposada en l'apartat 2.2.1 Justificació i restriccions de l'esquema relacional degut a la simplicitat del seu disseny.

2.2 Disseny lògic: Model relacional

L'etapa del disseny lògic pren com a punt de partida l'esquema conceptual, i la seva finalitat és transformar-ho en un altre model (estructura de dades) de manera que s'adapti a la tecnologia que s'emprarà posteriorment en la fase d'implementació de la BD. En aquesta fase es té en compte quin tipus de base de dades s'emprarà (relacional, jeràrquic, etc) i a més també contempla quin SGBD s'utilitzarà.

En aquest treball el model de dades que s'emprarà és el model relacional, i el SGBD serà Oracle. El model relacional proporciona una estructura de les dades per a representar la informació desitjada del món real, la qual es troba constituïda per un conjunt de relacions.

Hi ha un mètode que permet transformar el model E/R a model relacional mitjançant l'aplicació d'una sèrie de regles definides. La taula que es mostra a continuació resumeix les regles més rellevants associades a dita transformació.

Element del model ER	Transformació al model relacional
Entitat	Relació
Interrelació 1:1	Clau forana
Interrelació 1:N	Clau forana
Interrelació M:N	Relació
Interrelació n-ària	Relació
Interrelació recursiva	Com a les interrelacions no recursives: <ul style="list-style-type: none"> • Clau forana per a binàries 1:1 i 1:N • Relació per a binàries M:N i n-àries
Entitat dèbil	La clau forana de la interrelació identificadora forma part de la clau primària
Generalització/especialització	<ul style="list-style-type: none"> • Relació per a l'entitat superclasse • Relació per a cadascuna de les entitats subclasse
Entitat associativa	La transformació de la interrelació que l'origina és alhora la seva transformació

Així doncs, el resultat de la transformació dona pas a un diagrama compost per un conjunt de relacions amb els seus atributs, claus primàries i claus foranes. Observem que després de la transformació, el concepte de *interrelació* desapareix.

A continuació es mostrarà l'esquema lògic associat al sistema de gestió d'amonestacions i sancions en centres educatius i posteriorment es justificarà el disseny exposat mitjançant la segmentació d'aquest en parts amb les seves corresponents explicacions.

Model Relacional – Part1

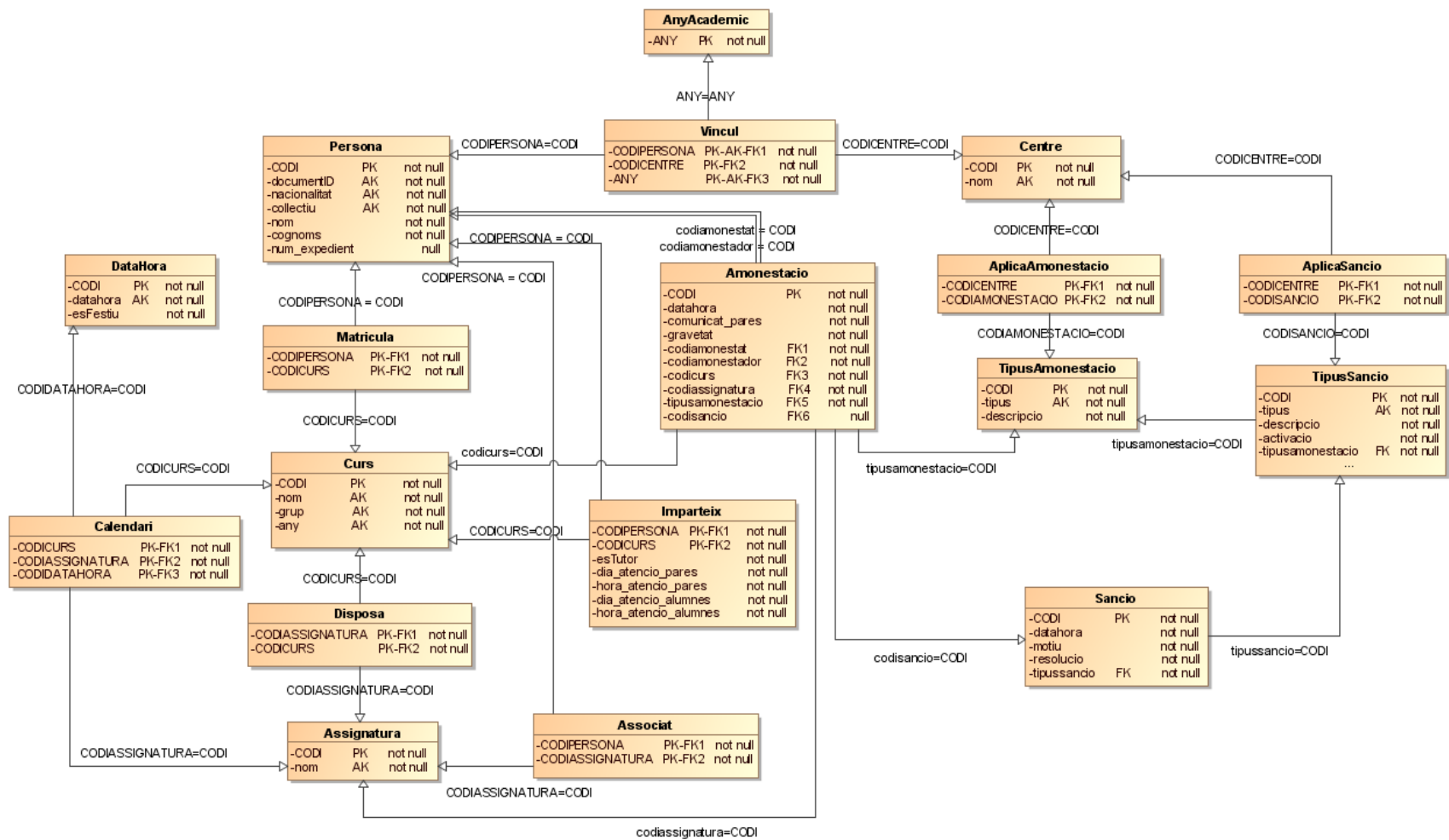


Figura 10

Model Relacional – Part2

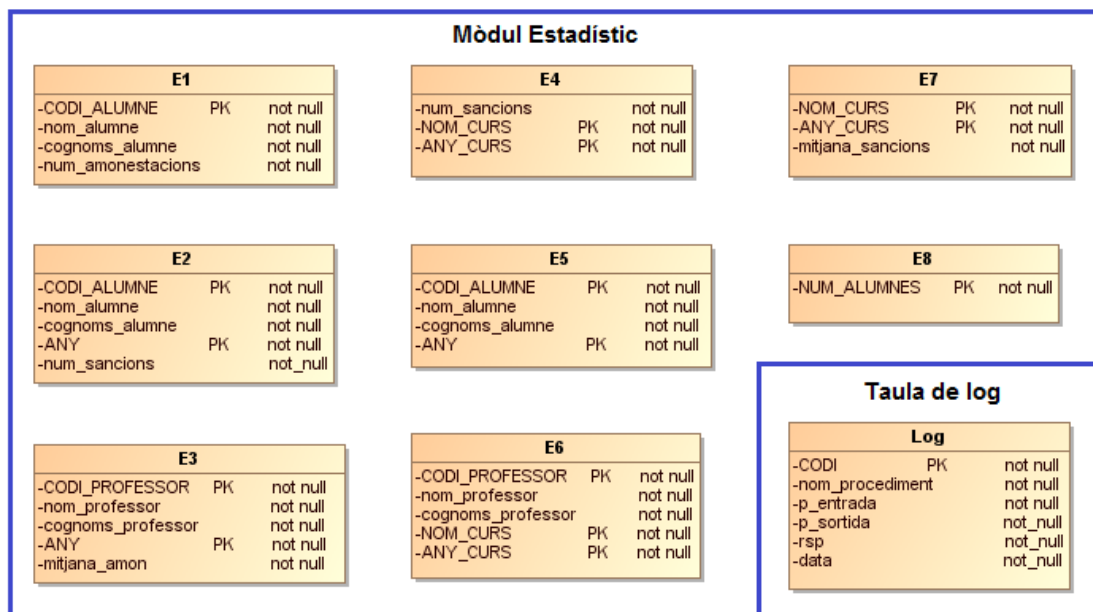


Figura 11

2.2.1 Justificació i restriccions de l'esquema relacional

De la mateixa manera que s'ha justificat el disseny conceptual, en aquest apartat s'argumentarà les decisions que s'han dut a terme durant l'elaboració del disseny lògic. També s'aprofitarà per posar de manifest els límits o restriccions del disseny proposat.

Amb l'objectiu de realitzar una explicació aclaridora i concisa, es procedirà de la mateixa forma que s'ha emprat per justificar el disseny conceptual, és a dir, fent una segmentació del disseny en parts amb les seves corresponents explicacions.

Aprofitant que l'element clau del model relacional és la taula, es dividirà l'esquema relacional anteriorment mostrat en taules independents. D'aquesta manera es podrà explicar per a que serveix una taula en concret, quin paper tenen els seus atributs, i quin domini i restriccions tenen aquests.

Finalment abans de començar amb la justificació del model, indicar que les taules que es mostraran segueixen aquesta estructuració:

<nom de la taula>				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
Atribut_1	Tipus_1	Domini_1	Descripció_1	Restricció_1
Atribut_2	Tipus_2	Domini_2	Descripció_2	Restricció_2
...
Atribut_N	Tipus_N	Domini_N	Descripció_N	Restricció_N

La BD del sistema de gestió d'amonestacions i sancions en centres educatius disposarà de les següents taules:

Persona				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(12)	Identificador únic que distingeix inequívocament a cada persona	NOT NULL
documentID	AK	VARCHAR2(20)	Pot ser el DNI o el passaport	NOT NULL
nacionalitat	AK	VARCHAR2(25)	Especifica la nacionalitat d'una persona	NOT NULL
collectiu	AK	VARCHAR2(20)	Indica el col·lectiu de la persona	NOT NULL i ('ALUMNE' o 'PROFESSOR' o ...)
nom	-	VARCHAR2(30)	Descriu el nom d'una persona	NOT NULL
cognoms	-	VARCHAR2(50)	Descriu els cognoms d'una persona	NOT NULL
num_expedient	-	NUMBER(12)	Especifica el número d'expedient (només alumnes)	NULL

Registre d'exemple que emmagatzema la taula:

CODI	documentID	nacionalitat	collectiu	nom	cognoms	num_expedient
26	78451278Q	ESPANYOLA	ALUMNE	MERCE	COSTA TORIBIO	52

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema la informació essencial relativa als professors i als alumnes de tots els centres educatius catalans.

Aspectes destacats del disseny:

- L'atribut *collectiu* té un paper clau i s'utilitza per diferenciar si una persona és alumne, professor, exalumne o exprofessor.
- Els atributs *CODI* i *num_expedient* s'autoincrementen automàticament quan es dona d'alta a una nova persona mitjançant l'ús del parell de components *sequence-trigger* definits per aquesta taula.
- L'atribut *num_expedient* pot ser NULL en el cas que la persona sigui un professor.
- S'ha definit una clau alternativa formada pels atributs *documentID*, *nacionalitat* i *collectiu*. L'objectiu d'aquesta és que el sistema pugui assumir el fet de que una determinada persona que ha estat donada d'alta com alumne, al cap d'uns anys pugui ser donada d'alta de nou amb un col·lectiu de professor.

Atributs opcionals descartats:

Es va prendre la decisió d'emmagatzemar només la informació relativa a les persones que realment fos purament imprescindible perquè el sistema d'amonestacions i sancions funcionés correctament.

Es podrien haver definit els següents atributs relatius a emmagatzemar les dades personals i de contacte:

Data de naixement, Ciutat de residència, Codi Postal, País de residència, Adreça residència, Telèfon de contacte, Email de contacte, Sexe, etc

Però donat a que són dades supèrflues per aquest sistema en concret, juntament amb el problema de temps i feina que suposaria emplenar aquests atributs a l'hora de poblar les taules, finalment es varen descartar.

Curs				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(12)	Identificador únic que distingeix inequívocament a cada curs	NOT NULL
nom	AK	VARCHAR2(25)	El nom del curs	NOT NULL
grup	AK	VARCHAR2(5)	Especifica el grup del curs	NOT NULL
any_	AK	NUMBER(4)	Indica l'any acadèmic del curs en qüestió	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODI	nom	grup	any_
38	TERCER ESO	A	2011

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema la informació de cadascun dels cursos que es realitzen en els centres educatius catalans.

Aspectes destacats del disseny:

- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta un nou curs a la base de dades mitjançant l'ús del parell de components *sequence-trigger* definits per aquesta taula.

Matricula				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODIPERSONA	PK, FK1	NUMBER(12)	Referència a Persona(CODI)	NOT NULL
CODICURS	PK, FK2	NUMBER(12)	Referència a Curs(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODIPERSONA	CODICURS
120	5

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema les diverses matricules de tots els estudiants. Associa un alumne amb un o varis cursos i viceversa.

Atributs opcionals descartats:

Es podria haver agregat un atribut que indiqués el cost de la matricula en qüestió i un altre que especificqués un identificador de la matricula, però en els requeriments funcionals del sistema no es contemplaven i per aquest fet no s'ha agregat cap atribut addicional.

Assignatura				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(8)	Identificador únic que distingeix inequívocament a l'assignatura	NOT NULL
nom	AK	VARCHAR2(40)	Especifica el nom de l'assignatura	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODI	nom
8	LITERATURA 1

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema el llistat d'assignatures que s'imparteixen en els centres educatius.

Aspectes destacats del disseny:

- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta una nova assignatura a la base de dades mitjançant l'ús del parell de components *sequence-trigger* definits per aquesta taula.
- L'atribut *nom* s'ha definit com a clau alternativa amb l'objectiu d'evitar que es puguin donar d'alta dues assignatures amb el mateix nom. Es suposa que la nomenclatura de les assignatures d'una mateixa temàtica segueix el format: Matemàtiques 1, Matemàtiques 2, etc

Atributs opcionals descartats:

Es podria haver definit un atribut que indiqués el nombre de crèdits de l'assignatura, però no aportava cap funcionalitat al sistema i no s'ha definit.

Disposa				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI ASSIGNATURA	PK, FK1	NUMBER(8)	Referència a Assignatura(CODI)	NOT NULL
CODICURS	PK, FK2	NUMBER(12)	Referència a Curs(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODIASSIGNATURA	CODICURS
21	3

Informació rellevant:

Finalitat de la taula: La taula conté la relació d'assignatures que s'imparteixen en un determinat curs, és a dir, associa una assignatura amb un o varis cursos i viceversa.

Centre				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(8)	Identificador únic que distingeix inequívocament al centre	NOT NULL
nom	AK	VARCHAR2(40)	Especifica el nom del centre	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODI	nom
5	I.E.S. BADALONA

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema el llistat de centres educatius d'ensenyament secundari catalans.

Aspectes destacats del disseny:

- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta un nou centre, mitjançant l'ús del parell de components *sequence-trigger* definits per aquesta taula.
- L'atribut *nom* s'ha definit com a clau alternativa amb l'objectiu d'evitar que es puguin donar d'alta dos centres amb el mateix nom.

Atributs opcionals descartats:

Es podria haver definit diversos atributs que aportessin informació addicional del centre, com per exemple: *La Ciutat*, *Codi postal*, *Adreça*, *Data d'obertura*, *Capacitat del centre*, etc. Però donat a que són dades supèrflues per aquest sistema en concret, finalment es varen descartar.

AnyAcademic				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
ANY_	PK	NUMBER(4)	Especifica l'any acadèmic d'un curs	NOT NULL

Registre d'exemple que emmagatzema la taula:

ANY_
2011

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema els anys acadèmics a partir dels quals es desitja tenir l'històric d'informació del sistema de gestió d'amonestacions i sancions en centres educatius.

Vincul				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODIPERSONA	PK, AK, FK1	NUMBER(12)	Referència a Persona(CODI)	NOT NULL
CODICENTRE	PK, FK2	NUMBER(8)	Referència a Centre(CODI)	NOT NULL
ANY_	PK, AK, FK3	NUMBER(4)	Referència a AnyAcademic(ANY_)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODIPERSONA	CODICENTRE	ANY_
8	10	2010

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema la vinculació que ha tingut una persona amb un centre en un determinat any. És a dir, emmagatzema l'històric de cadascuna de les persones que han estudiat o treballat en un centre a una determinada època.

Aspectes destacats del disseny:

- Els atributs *CODIPERSONA* i *ANY_* s'han definit com a clau alternativa amb l'objectiu d'evitar que una persona en un mateix any estudií o treballi en dos centres alhora.

Imparteix				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODIPERSONA	PK, FK1	NUMBER(12)	Referència a Persona(CODI)	NOT NULL
CODICURS	PK, FK2	NUMBER(12)	Referència a Curs(CODI)	NOT NULL
esTutor	-	CHAR(1)	Indica si el professor és el tutor del curs	NOT NULL i 'S' o 'N'
dia_atencio_pares	-	VARCHAR2(10)	Especifica el dia d'atenció als pares	NOT NULL
hora_atencio_pares	-	VARCHAR2(2)	Especifica l'hora d'atenció als pares	NOT NULL
dia_atencio_alumnes	-	VARCHAR2(10)	Especifica el dia d'atenció als alumnes	NOT NULL
hora_atencio_alumnes	-	VARCHAR2(2)	Especifica l'hora d'atenció als alumnes	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODI PERSONA	CODICURS	esTutor	dia_atencio_pares	hora_atencio_pares	dia_atencio_alumnes	hora_atencio_alumnes
158	5	S	DILLUNS	18	DIMECRES	14

Informació rellevant:

Finalitat de la taula: Emmagatzemar la informació relativa a la vinculació dels professors amb els diferents cursos. És a dir, associa un professor amb un o varis cursos i viceversa.

Aspectes destacats del disseny:

- L'atribut *esTutor* només pot prendre dos valors, S o N (se n'espera un comportament booleà)

Associat				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODIPERSONA	PK, FK1	NUMBER(12)	Referència a Persona(CODI)	NOT NULL
CODI ASSIGNATURA	PK, FK2	NUMBER(8)	Referència a Assignatura(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODIPERSONA	CODIASSIGNATURA
13	2

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema la relació de cadascun dels professors amb les assignatures que imparteixen. És a dir, associa un professor amb una o varies assignatures i viceversa.

DataHora				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(15)	Identificador únic que distingeix inequívocament la data i hora	NOT NULL
datahora	AK	DATE	Especifica una data i hora	NOT NULL
esFestiu	-	CHAR(1)	Indica si el dia és festiu	NOT NULL i 'S' o 'N'

Registre d'exemple que emmagatzema la taula:

CODI	datahora	esFestiu
123	31/03/2011 11	N

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema tots els dies d'un any acadèmic amb les seves respectives hores.

Aspectes destacats del disseny:

- L'atribut *esFestiu* només pot prendre dos valors, S o N (se n'espera un comportament booleà). Serveix per contemplar els dies festius definits per la conselleria d'educació.
- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta un dia i hora, mitjançant l'ús del parell de components *sequence-trigger* definits per aquesta taula.

Calendari				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODICURS	PK, FK1	NUMBER(12)	Referència a Curs(CODI)	NOT NULL
CODI ASSIGNATURA	PK, FK2	NUMBER(8)	Referència a Assignatura(CODI)	NOT NULL
CODIDATAHORA	PK, FK3	NUMBER(15)	Referència a DataHora(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODICURS	CODIASSIGNATURA	CODIDATAHORA
12	20	15

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema el calendari dels diferents cursos. Per aquest motiu, la taula *calendari* relaciona un curs, amb una o més assignatures i l'horari d'aquestes.

TipusAmonestacio				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(8)	Identificador únic que distingeix inequívocament el tipus d'amonestació	NOT NULL
tipus	AK	VARCHAR2(25)	Indica el tipus d'amonestació	NOT NULL
descripcio	-	VARCHAR2(200)	Defineix a que es deu l'amonestació	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODI	tipus	descripcio
3	alumne_malparlat	L'alumne ha sigut malparlat

Informació rellevant:

Finalitat de la taula: Aquesta taula conté la informació dels tipus d'amonestacions establertes en els centres educatius.

Aspectes destacats del disseny:

- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta un nou tipus d'amonestació, mitjançant l'ús d'un parell de components *sequence-trigger*.

AplicaAmonestacio				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODICENTRE	PK, FK1	NUMBER(8)	Referència a Centre(CODI)	NOT NULL
CODI AMONESTACIO	PK, FK2	NUMBER(8)	Referència a TipusAmonestacio(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODICENTRE	CODIAMONESTACIO
7	3

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema la relació del tipus d'amonestacions que s'apliquen a cadascun dels centres. És a dir, associa un centre amb un o varis tipus d'amonestacions i viceversa.

TipusSancio				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(8)	Identificador únic que distingeix inequívocament el tipus de sanció	NOT NULL
tipus	-	VARCHAR2(25)	Indica el tipus de sanció	NOT NULL
descripcio	-	VARCHAR2(200)	Defineix a que es deu la sanció	NOT NULL
activacio	-	NUMBER(4)	Defineix la regla per activar la sanció automàtica	NOT NULL
tipusamonestacio	FK	NUMBER(8)	Referència a TipusAmonestacio(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODI	tipus	descripcio	activacio	tipusamonestacio
2	prova_extra	Fer una prova d'avaluació extra	5	2

Informació rellevant:

Finalitat de la taula: Aquesta taula conté la informació dels tipus de sancions establertes en els centres educatius.

Aspectes destacats del disseny:

- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta un nou tipus de sanció, mitjançant l'ús del parell de components *sequence-trigger* definits a la taula.
- L'atribut *activacio* s'empra per definir les regles per activar sancions automàtiques.
- La clau forana *tipusamonestacio* serveix per especificar que cada tipus de sanció ha d'anar lligat a un tipus d'amonestació definit.

AplicaSancio				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODICENTRE	PK, FK1	NUMBER(8)	Referència a Centre(CODI)	NOT NULL
CODISANCIO	PK, FK2	NUMBER(8)	Referència a TipusSancio(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODICENTRE	CODISANCIO
6	2

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema la relació del tipus de sancions que s'apliquen a cadascun dels centres. És a dir, associa un centre amb un o varis tipus de sancions i viceversa.

Sancio				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(10)	Identificador únic que distingeix inequívocament una sanció	NOT NULL
datahora	-	DATE	Desa la data de llançament de la sanció	NOT NULL
motiu	-	VARCHAR2(300)	Indica el motiu de la sanció	NOT NULL
resolucio	-	VARCHAR2(300)	Especifica la resolució de la sanció	NOT NULL
tipussancio	FK	NUMBER(8)	Referència a TipusSancio(CODI)	NOT NULL

Registre d'exemple que emmagatzema la taula:

CODI	datahora	motiu	resolucio	tipussancio
1	6/04/2011 13	L'alumne ha fet soroll 5 vegades	Se li comunica a l'alumne que la propera setmana haurà de fer una prova extra	2

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema les sancions imposades als alumnes de tots els centres educatius catalans.

Aspectes destacats del disseny:

- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta una nova sanció, mitjançant l'ús del parell de components *sequence-trigger* definits a la taula.
- La clau forana *tipussancio* serveix per especificar que cada sanció ha de ser d'un dels tipus de sancions definits.

Amonestacio				
Nom de l'atribut	Tipus d'atribut	Domini de l'atribut	Descripció de l'atribut	Restriccions
CODI	PK	NUMBER(10)	Identificador únic que distingeix inequívocament l'amonestació	NOT NULL
datahora	-	DATE	Data i hora de l'amonestació	NOT NULL
comunicat_pares	-	CHAR(1)	Indica si s'ha comunicat als pares	NOT NULL i 'S' o 'N'
gravetat	-	VARCHAR2(8)	Especifica la gravetat	NOT NULL i ('NORMAL' o 'ALTA')
codiamonestat	FK1	NUMBER(12)	Referència a Persona(CODI)	NOT NULL
codiamonestador	FK2	NUMBER(12)	Referència a Persona(CODI)	NOT NULL
codicurs	FK3	NUMBER(12)	Referència a Curs(CODI)	NOT NULL
codiassignatura	FK4	NUMBER(8)	Referència a Assignatura(CODI)	NOT NULL
tipusamonestacio	FK5	NUMBER(8)	Referència a TipusAmonestacio(CODI)	NOT NULL
codisancio	FK6	NUMBER(10)	Referència a Sancio(CODI)	NULL

Registre d'exemple que emmagatzema la taula:

CODI	datahora	comunicat_pares	gravetat	codiamonestat
78	28/04/2008 11	S	NORMAL	12
codi amonestador	codicurs	codi assignatura	tipus amonestacio	codisancio
121	5	14	3	1

Informació rellevant:

Finalitat de la taula: Aquesta taula emmagatzema les amonestacions imposades als alumnes de tots els centres educatius catalans.

Aspectes destacats del disseny:

- L'atribut *CODI* s'autoincrementa automàticament quan es dona d'alta una nova amonestació, mitjançant l'ús del parell de components *sequence-trigger* definits a la taula.
- L'atribut *comunicat_pares* només pot prendre dos valors, S o N (se n'espera un comportament booleà).
- L'atribut *gravetat* només pot prendre dos valors, NORMAL o ALTA.
- La clau forana *codisancio* pot prendre el valor NULL. Aquest atribut serveix per especificar si l'amonestació comporta una sanció.

Finalment només resten les taules corresponents al mòdul estadístic i la taula de log. Per una qüestió d'espai, no es mostrarà de manera gràfica l'estructura d'aquestes, al contrari de com s'ha procedit fins ara amb les demés taules del disseny. A continuació s'explicarà de manera clara i concisa l'estructura de dites taules i es justificarà el seu disseny.

El mòdul estadístic estarà format per vuit taules independents, cadascuna de les quals contindrà de manera actualitzada la informació corresponent a cadascuna de les vuit consultes especificades als requeriments funcionals descrits pel client.

El nom d'aquestes taules seguirà la nomenclatura "E + número de consulta que respon". És a dir, que la taula amb el nom *E1* conté la informació estadística que dona resposta a la consulta: 1-El número d'amonestacions per alumne (independentment del curs).

Internament l'estructura de dites taules és molt senzilla donat que només s'ha emprat atributs bàsics, sense recorre a l'ús de claus foranes o artificials. Es pot veure l'estructura exacta d'aquestes a la *Figura 11* o al final del fitxer "*Creacio de taules.sql*".

Per finalitzar amb el mòdul estadístic, indicar que més endavant al capítol de la implementació de la BD s'explicarà com s'omplen aquestes taules.

La taula de log té la mateixa estructura que qualsevol de les taules que componen el mòdul estadístic. És a dir, es tracta d'una taula que no té claus foranes i la única peculiaritat destacable és que la seva clau primària fa ús d'un parell de components *sequence-trigger* que fan autoincrementar la clau cada cop que s'insereix un nou registre.

La resta d'atributs que componen dita taula són els següents:

- *nom_procediment* : Conté el nom del procediment que s'ha executat en la BD.
- *p_entrada* : Conté el nom i els valors de cadascun dels paràmetres d'entrada del procediment que s'ha cridat.
- *p_sortida* : Conté el nom i els valors de cadascun dels paràmetres de sortida del procediment que s'ha cridat. (finalment tots els procediments d'aquest projecte tenen un únic paràmetre de sortida, el *rsp*)
- *rsp* : Conté el nom i el valor del paràmetre de sortida RSP. Bàsicament el valor serà `OK` si el procediment ha dut a terme la seva funció correctament, en cas contrari especificarà l'error i el tipus d'error produït durant l'execució d'aquest.
- *datahora* : Conté el dia, l'hora, els minuts i els segons exactes de quan s'ha produït l'execució del procediment.

Al capítol corresponent a la implementació de la BD s'explicarà com i quan s'omple la taula de log.

2.3 Disseny físic

L'etapa del disseny físic pren com a punt de partida l'estructura obtinguda a l'etapa del disseny lògic, i el seu objectiu principal és implementar una estructura física o interna que pugui suportar el model lògic amb la màxima eficiència en termes de recursos de hardware.

Aquesta etapa depèn totalment del SGBD que s'utilitzarà, donat que cadascun té les seves característiques i poden implementar diferents solucions respecte l'emmagatzemament físic de les dades.

Els aspectes que cal contemplar bàsicament consisteixen en l'elecció d'estructures físiques d'implementació de les relacions, la selecció de la mida de les memòries intermèdies o *buffers*, la mida de les pàgines, etc.

A més es té en compte quines dades són les que es consulten més habitualment, quin volum de dades s'espera gestionar i quin temps de resposta es considera acceptable en cada cas.

En definitiva, els objectius del disseny físic són:

- Minimitzar el temps de resposta a les consultes.
- Optimitzar l'espai que les dades ocupen al disc d'emmagatzematge.
- Garantir l'emmagatzemament persistent i segur de les dades.
- Optimitzar els recursos del sistema en termes de hardware (CPU, RAM, busos, etc.)

Per aconseguir els objectius descrits, els principals mecanismes o components que utilitzen els SGBD de l'actualitat són:

- Índexs secundaris
- Registres físics
- Punters
- Agrupaments (cluster)
- Bloqueig i comprensió de dades
- Adreçament calculat (Hashing)
- Assignació d'espais d'emmagatzematge com a memòries intermèdies (buffers)
- Assignació de conjunts de dades a particions i a dispositius físics

En relació al sistema de gestió d'amonestacions i sancions en centres educatius, després de consultar al client, es decideix que no és necessari aprofundir en els aspectes relacionats amb l'optimització de consultes.

Així doncs, l'únic component d'optimització que s'emprarà serà l'ús dels índexs que el SGBD Oracle crea per defecte quan es defineix la clau primària de cadascuna de les taules, tal com mostra la següent imatge:

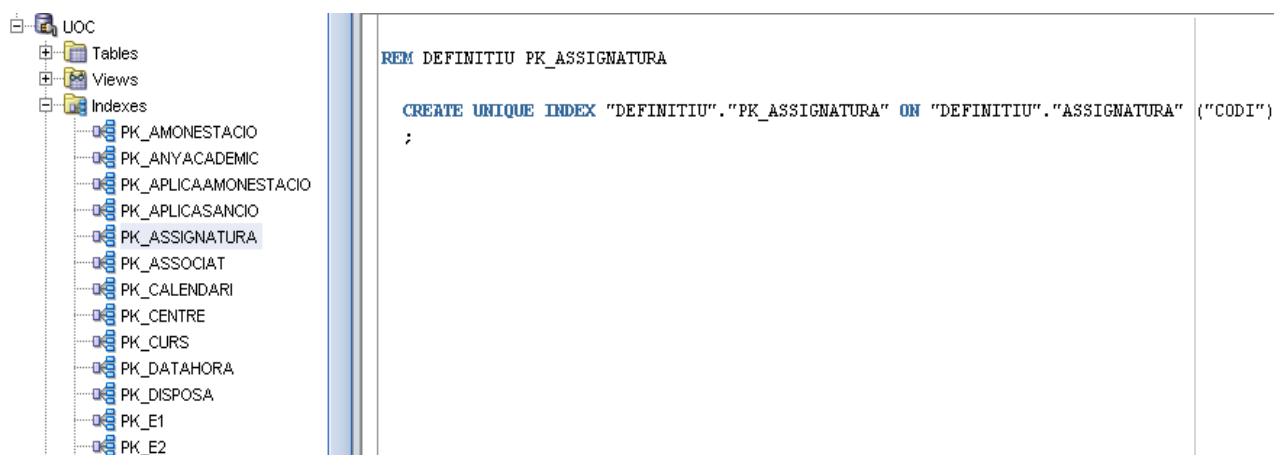


Figura 12

3. Implementació de la Base de Dades

Un cop definit completament el disseny de la BD, es pot procedir a la seva implementació. Abans de tractar els aspectes més tècnics i especificar les solucions aportades per realitzar cada funcionalitat del sistema de gestió d'amonestacions i sancions, es descriurà la metodologia de treball emprada.

La fase d'implementació s'ha completat seguint en ordre les següents passes:

- 1- S'ha creat el script "*Creacio de taules.sql*" per crear totes les taules, triggers i seqüències vinculades a les claus artificials emprades per aquestes taules. Aquest procés ha consistit en agafar les taules descrites anteriorment en el disseny lògic, i codificar-les d'una en una amb les sentències del llenguatge de programació d'Oracle.
- 2- Posteriorment, s'ha creat el script "*Carrega de dades.sql*" per omplir totes les taules i comprovar el seu funcionament. Aquest fitxer al principi només era constituït per sentències INSERT, però un cop es va finalitzar els procediments d'alta dels diferents objectes, va ser modificat perquè en comptes d'utilitzar només les sentències INSERT utilitzés els procediments emmagatzemats.
- 3- D'un en un es varen crear els fitxers d'alta, baixa i modificació de les diferents entitats que sol·licitaven els requeriments funcionals. Quan va ser el torn de codificar els procediments d'ABM d'amonestacions, es va procedir concurrentment a crear el fitxer "*Triggers Gestio Sancions Automatiques.sql*" el qual es troba format per quatre disparadors que implementen l'aplicació de sancions automàtiques. Aquest fitxer ha sigut la part més complexa d'implementar.
- 4- A continuació es va implementar el fitxer "*Procediments de consulta.sql*" per crear els procediments que donen resposta a les quatre consultes sol·licitades.
- 5- En acabar tots els procediments anteriors, va ser el torn de generar el fitxer "*Gestio Modul Estadistic.sql*" per crear els procediments de gestió del Mòdul Estadístic.
- 6- Finalment la fase d'implementació va culminar amb la realització del fitxer "*Joc de proves i errors.sql*" que va ser emprat per realitzar proves i crear situacions d'excepció.

Cada script es provava exhaustivament un cop s'havia realitzat, evitant d'aquesta manera que s'acumulés una gran quantitat de funcionalitats per testejar a la fase de proves. Gràcies a aquest fet, la pèrdua de temps per trobar i corregir errors ha sigut mínima.

També indicar que cada fitxer s'anava documentant mitjançant comentaris interns a mesura que s'anava programant. Aquest fet és important donat que passats uns dies si es requeria modificar el fitxer, els comentaris ajudaven d'immediat a saber que feia aquell conjunt d'instruccions.

Els dubtes que anaven sorgint a mesura que s'anava avançant en la implementació, van ser consultats amb el client, que ràpidament va indicar que s'esperava de cada funcionalitat.

A continuació es descriurà amb detall com funcionen cadascun dels fitxers resultants de l'elaboració del producte. La descripció serà des d'un punt de vista d'alt nivell, donat que si es desitja sapiguer algun aspecte molt concret d'implementació, es recomana obrir el fitxer en qüestió, on línea a línea es podrà trobar comentaris explicatius que donen resposta a tots els aspectes de la implementació.

Amb l'objectiu de realitzar una explicació aclaridora, es procedirà primer a comentar els aspectes o patrons generals que tenen en comú tots els scripts i posteriorment s'entrarà en les particularitats de cadascun d'ells.

3.1 Aspectes generals dels scripts

Tots els scripts s'han desenvolupat seguint els mateixos patrons i metodologia, d'aquesta manera s'ha aconseguit homogeneïtzar totalment el sistema. Això és un gran avantatge donat que permet que l'aplicació pugui ser mantinguda i millorada posteriorment amb facilitat pels desenvolupadors, seguint una programació molt lògica i gens enrevessada.

A continuació s'explicarà aquests aspectes comuns.

3.1.1 Capçalera descriptiva

A la part superior de cada fitxer trobarem una capçalera amb informació relativa al fragment de codi que hi ha immediatament a sota d'aquesta. La capçalera descriu que fa el procediment o fragment de codi a alt nivell, especifica i exemplifica els tipus i valors possibles per a cada paràmetre d'entrada i de sortida, indica quin són els valors de retorn que s'esperen després de la seva execució i per últim mostra un exemple de crida per iniciar la seva execució.

De manera gràfica, una capçalera tindrà aquest aspecte:

```

-----
-- Script per crear els procediments d'ABM dels alumnes
-----

/*****
Nom procediment:          alta_alumne

Funcionalitat:           Gestiona l'alta d'un alumne a la taula Persona de la BD

Paràmetres d'entrada:    (ins_documentID, ins_nacionalitat, ins_nom, ins_cognoms)
-----
NOM                       TIPUS          VALOR EXEMPLE
ins_documentID            string      98421789G
ins_nacionalitat          string      ESPANYOLA
ins_nom                    string      XAVIER
ins_cognoms                string      MIRANDA CANTERO

Paràmetres de sortida:   (rsp)
-----
NOM                       TIPUS          VALOR EXEMPLE
rsp                        string         OK

Valor de retorn:         La sortida serà 'OK' si l'alta s'ha realitzat,
                          en cas contrari retorna: 'ERROR: Tipus d'error'

Exemple d'ús:            alta_alumne('98421789G', 'ESPANYOLA', 'XAVIER', 'MIRANDA CANTERO', rsp);
*****/

```

Figura 13

3.1.2 Tractament de les dades d'entrada

Tots els procediments que treballen amb paràmetres d'entrada de tipus cadena de caràcters o *string* disposen d'un tractament que converteix a majúscules totes aquestes dades i a més si per error s'havia introduït algun espai en blanc a l'inici o al final de la cadena, el treu.

Tot i que cap requeriment funcional ho especificava, aquest tractament de dades té l'avantatge de quan es realitzi una consulta referenciant aquestes dades no hi hagi problemes de tipus *case sensitive*. És a dir, que una consulta retorni que no existeix cap valor quan realment si que existeix, però no el troba donat que no coincideixen majúscules i minúscules. Així doncs, es recomana emprar majúscules quan es desitgi realitzar alguna consulta i aquesta utilitzi algun *string*.

Les dades de sortida també són tractades de la mateixa manera, però com aquestes no depenen de l'usuari, no cal comentar-les.

A nivell visual, en els procediments serà present un fragment de codi com aquest:

```
-- Es filtra les dades d'entrada: Es converteix a majúscules totes les cadenes de caràcters i s'eliminen
-- els possibles espais del principi i final
i_nom := UPPER(TRIM(ins_nom));
i_grup := UPPER(TRIM(ins_grup));
i_any := UPPER(TRIM(ins_any));
```

Figura 14

3.1.3 Tractament d'excepcions

El tractament d'excepcions és un factor fonamental per mantenir la integritat de les dades al sistema. També permet sapiguer si una transacció o conjunt d'instrucció ha donat el resultat esperat o si s'ha produït un error, sapiguer el motiu.

Les excepcions es poden definir o bé de manera manual o bé automàtiques. Com a manual s'entén que el programador crea una regla específica (que bàsicament és formada per una sentència condicional *if*) perquè quan es donin unes determinades circumstàncies es consideri un cas d'error i es procedeixi a fer les tasques oportunes (mitjançant la crida *RAISE*) per tractar-ho.

Les automàtiques s'entenen com aquelles regles comuns e inherents a tota base de dades relacional que es tenen que complir. Per exemple, la no duplicat d'una clau primària, l'existència dels valors que referencien les claus foranes, etc.

Aquestes excepcions quan es produeixen, es pot captar l'error específic i desar-ho com a text gràcies a la funció *SQLERRM*, la qual torna el missatge d'error associat a l'excepció produïda més recentment.

En el sistema de gestió d'amonestacions i sancions, quan una determinada instància que s'executa en qualsevol dels procediments passa tots els possibles filtres d'error sense generar cap excepció, aleshores es realitza un *COMMIT* que confirma la transacció desant de manera persistent els canvis.

En el cas que es produeixi algun error, aquest serà tractat en l'apartat *EXCEPTION* del procediment. El tractament consistirà en obtenir el motiu que ha produït l'error i realitzar un *ROLLBACK* desfent els canvis produïts a l'última transacció.

De manera gràfica, la detecció d'excepcions tindrà el següent aspecte:

```
-- Comprovacions dels possibles errors
-- Comprovació perquè els atributs definits com a NOT NULL siguin respectats al inserir
IF (ins_codi_curs IS NULL) OR (ins_codi_assignatura IS NULL) OR (ins_codi_datahora IS NULL) THEN
    RAISE camps_null;
END IF;

-- Com tots els atributs de Calendari són claus foranes, primer s'ha de comprovar que les tuples que referencien existeixin.
-- Es verifica que existeixi el curs amb el codi introduït.
SELECT COUNT (*)
    INTO i_codi_curs
    FROM Curs
    WHERE (CODI = ins_codi_curs);
-- Si la cerca anterior no té èxit, es llança una excepció.
IF (i_codi_curs = 0) THEN
    RAISE noexisteixcurs;
END IF;

-- Es verifica que existeixi l'assignatura amb el codi introduït.
SELECT COUNT (*)
    INTO i_codi_assignatura
    FROM Assignatura
    WHERE (CODI = ins_codi_assignatura);
-- Si la cerca anterior no té èxit, es llança una excepció.
IF (i_codi_assignatura = 0) THEN
    RAISE noexisteixassignig;
END IF;
```

Figura 15

I el tractament de les excepcions seguirà aquesta estructura:

```
EXCEPTION
-- Error degut a no respectar els atributs NOT NULL
WHEN camps_null THEN
    -- Es desfan els canvis produïts a la transacció
    ROLLBACK;
    -- Es registra la crida a la taula Log
    rsp := 'ERROR: Hi ha camps amb valor NULL que no ho poden ser.';
    p_sortida := 'rsp: ' || rsp;
    INSERT INTO Log VALUES ('', 'alta_calendari', p_entrada, p_sortida, rsp, to_char(sysdate, 'DD/MM/YYYY HH24:MI:SS'));
    DBMS_OUTPUT.PUT_LINE(rsp);
    COMMIT;
-- Error degut a que no existeix cap Curs amb el codi especificat
WHEN noexisteixcurs THEN
    -- Es desfan els canvis produïts a la transacció
    ROLLBACK;
    -- Es registra la crida a la taula Log
    rsp := 'ERROR: Amb el codi especificat, no existeix cap Curs.';
    p_sortida := 'rsp: ' || rsp;
    INSERT INTO Log VALUES ('', 'alta_calendari', p_entrada, p_sortida, rsp, to_char(sysdate, 'DD/MM/YYYY HH24:MI:SS'));
    DBMS_OUTPUT.PUT_LINE(rsp);
    COMMIT;
-- Errors genèrics
WHEN OTHERS THEN
    -- Es desfan els canvis produïts a la transacció
    ROLLBACK;
    -- Es registra la crida a la taula Log
    v_errm := SUBSTR(SQLERRM, 1, 200);
    rsp := 'ERROR: ' || v_errm;
    p_sortida := 'rsp: ' || rsp;
    INSERT INTO Log VALUES ('', 'alta_calendari', p_entrada, p_sortida, rsp, to_char(sysdate, 'DD/MM/YYYY HH24:MI:SS'));
    DBMS_OUTPUT.PUT_LINE(rsp);
    COMMIT;
```

Figura 16

3.1.4 Emmagatzemament dels registres de log

Uns dels requeriments de metodologia sol·licitats pel client consistia en desar totes les crides a procediments que es realitzessin.

Per implementar aquest requeriment s'ha optat per inserir el registre a la taula log un cop el procediment s'ha executat amb èxit, o bé si s'ha produït algun error, un cop aquest hagi sigut tractat.

Abans de poder enregistrar l'execució del procediment a la taula de log, s'han de preparar els valors a inserir. Bàsicament són tres valors a preparar: els paràmetres d'entrada, els de sortida i el valor rsp.

Els paràmetres d'entrada es desen en una variable de tipus string anomenada *p_entrada*.
Els paràmetres de sortida es desen en una variable de tipus string anomenada *p_sortida*.
El *rsp* es desa en una variable que té el seu mateix nom.

La resta d'atributs a inserir són el nom del procediment (que depèn de cada procediment que s'executi) i la data i hora de l'execució, que s'implementa amb la funció *sysdate*.

A continuació es mostra una captura que exemplifica el registre de logs:

```
-- Es desa l'entrada de dades sense alterar que s'ha realitzat mitjançant els paràmetres d'entrada
p_entrada := 'nom: ' || ins_nom;

-- Es registra la crida a la taula Log
rsp := 'OK';
p_sortida := 'rsp: ' || rsp;
INSERT INTO Log VALUES ('', 'alta_assignatura', p_entrada, p_sortida, rsp, to_char(sysdate, 'DD/MM/YYYY HH24:MI:SS'));
-- Es mostra per pantalla el resultat de l'operació
DBMS_OUTPUT.PUT_LINE(rsp);
-- Es confirmen els canvis produïts a la transacció
COMMIT;
```

Figura 17

Finalment després de la inserció reiterativa de registres, la taula *Log* presentarà un aspecte com aquest:

CODI	NOM_PROCEDIMENT	P_ENTRADA	P_SORTIDA	RSP	DATAHORA
1687	alta_curs	nom: SEGON ESO grup: H any: 2009	rsp: OK	OK	05/06/2011 11:19:43
1688	alta_curs	nom: SEGON ESO grup: H any: 2009	rsp: ERROR: ORA-00001: restricció única (DEFINITIU.LIN_CURS) violada	ERROR: ORA...	05/06/2011 11:19:43
1689	alta_assignatura	nom: FILOSOFIA	rsp: OK	OK	05/06/2011 11:19:43
1690	alta_assignatura	nom: FILOSOFIA	rsp: ERROR: ORA-00001: restricció única (DEFINITIU.LIN_ASSIGNATUR...	ERROR: ORA...	05/06/2011 11:19:43
1691	alta_calendari	Codi Curs: 1001 Codi Assignatura: 3 Codi Datahora: 1	rsp: ERROR: Amb el codi especificat, no existeix cap Curs.	ERROR: Amb ...	05/06/2011 11:19:43
1692	alta_tipus_amonestacio	Tipus: Descripció: L'alumne no ha fet els deures	rsp: ERROR: Hi ha camps amb valor NULL que no ho poden ser.	ERROR: Hi ha ...	05/06/2011 11:19:43
1693	alta_tipus_sancio	Tipus: prova_extra Descripció: Fer una prova d'avaluació extr...	rsp: ERROR: ORA-00001: restricció única (DEFINITIU.LIN_TIPUSSANCIO...	ERROR: ORA...	05/06/2011 11:19:43
1694	alta_amonestacio	Data i Hora: 21/09/10 Comunicat pares: N Gravetat: NORMAL...	rsp: ERROR: Amb el codi de l'amonestat especificat, no existeix cap pers...	ERROR: Amb ...	05/06/2011 11:19:43
1695	alta_amonestacio	Data i Hora: 30/10/08 Comunicat pares: S Gravetat: NORMAL...	rsp: ERROR: Amb el codi especificat, no existeix cap tipus d'amonestació.	ERROR: Amb ...	05/06/2011 11:19:43
1696	listar_amonestacions	No hi ha.	rsp: OK	OK	05/06/2011 11:33:57
1697	listar_alumnes	Codi Curs: 1	rsp: OK	OK	05/06/2011 11:34:21
1698	listar_amon_i_sancions	Codi Centre: 4	rsp: OK	OK	05/06/2011 11:34:41
1699	listar_amon_i_san_alumne	Codi Alumne: 2	rsp: OK	OK	05/06/2011 11:34:58

Figura 18

3.2 Aspectes específics dels scripts

Un cop definits els aspectes comuns que comparteixen tots els scripts, es procedirà a explicar les particularitats i limitacions de cadascun. La manera triada per fer-ho, serà comentar a alt nivell cada fitxer del producte.

Així doncs, aquest apartat donarà resposta a *què fa cada fitxer* i *com ho fa*, però sense entrar en detalls de programació excessivament concrets, els quals estan extensament comentats i argumentats en els mateixos arxius.

3.2.1 Script de creació de taules

L'arxiu "*Creacio de taules.sql*" és l'encarregat de crear totes les taules, triggers i seqüències vinculades a les claus artificials emprades per aquestes taules. Totes les taules segueixen exactament la mateixa estructura que l'especificada a l'esquema lògic.

S'ha optat per utilitzar una clau artificial autoincrementada anomenada *CODI* en la majoria de taules. Això s'ha determinat així per comoditat, donat que és molt pràctic referenciar a un registre per una clau primària formada per un únic atribut que per un conjunt.

En aquest fitxer també trobarem el paquet o *package* anomenat *common_pkg*, el qual es troba constituït per una sèrie de variables globals que s'utilitzen en els triggers i procediments que gestionen el sistema de llançament de sancions automàtiques i que més endavant s'explicaran.

El mateix fitxer conté també les instruccions de creació de la taula log i de les vuit taules del mòdul estadístic. En resum, aquest fitxer crea 27 taules que seran totes les necessàries per emmagatzemar la informació del sistema d'amonestacions i sancions.

3.2.2 Script de càrrega de dades

L'arxiu "*Carrega de dades.sql*" serveix per omplir totes les taules amb un joc de proves de mida considerable.

Aquest fitxer utilitza crides als diferents procediments d'alta (que a continuació s'explicaran) amb l'objectiu de poblar totes les taules. Quan no existeix el procediment d'alta per una taula en concret, aleshores s'empra una sentència de *INSERT* tradicional.

Si les taules són creades amb l'arxiu *Creacio de taules.sql* i no s'ha fet cap tipus d'alteració de l'estructura de les taules ni cap inserció de dades anterior, aleshores el script *Carrega de dades.sql* garanteix que el joc de proves sigui net i insereix absolutament totes les dades del script sense produir cap error.

A continuació es mostra una taula que especifica el número de dades inserides per aquest script a cadascuna de les taules de la base de dades:

Nom de la taula	Número de registres	Nom de la taula	Número de registres
Persona (alumnes)	110	Imparteix	270
Persona (professors)	50	Associat	100
Curs	60	DataHora	465
Matricula	230	Calendari	590
Assignatura	60	TipusAmonestacio	5
Disposa	270	AplicaAmonestacio	60
Centre	20	TipusSancio	5
AnyAcademic	5	AplicaSancio	33
Vincul	255	Amonestacio	121

3.2.3 Procediments d'ABM d'alumnes i professors

Amb els fitxers “*ABM alumne.sql*” i “*ABM professor.sql*” es crea a la BD els procediments d'alta, baixa i modificació dels alumnes i dels professors, respectivament.

Es tracta de scripts gairebé idèntics, i per aquest motiu s'expliquen conjuntament. Es descriurà la seva funcionalitat mitjançant una comparació entre ambdós fitxers.

Quan es dona d'**alta** un alumne a la base de dades, el registre corresponent tindrà el valor *ALUMNE* per l'atribut *collectiu*. En canvi l'alta d'un professor implica que el seu col·lectiu tingui el valor de *PROFESSOR*.

La resta d'atributs són idèntics, exceptuant l'atribut *num_expedient*. En el cas dels alumnes, el número d'expedient s'anirà autoincrementant a mesura que s'insereixen els estudiants. En el cas dels professors l'atribut serà *NULL* sempre.

Quan es dona de **baixa** un alumne o un professor, la seva informació relativa no s'elimina del sistema. La baixa s'ha considerat com un canvi de col·lectiu amb l'objectiu de poder mantenir un històric en el sistema. Així doncs una baixa d'alumne fa que aquest passi de tenir el col·lectiu *ALUMNE* a *EXALUMNE*. De la mateixa manera, un professor passarà de ser *PROFESSOR* a *EXPROFESSOR*.

Respecte a la **modificació** d'un alumne o un professor, no hi ha cap aspecte transcendent. El procediment es limita a sol·licitar el codi del registre a modificar i les noves dades que es desitgen emmagatzemar. En el cas que les noves dades provoquin una excepció, es mostra per pantalla l'error.

3.2.4 Procediments d'ABM dels cursos, assignatures i calendari escolar

Es comentarà la funcionalitat i característiques d'aquests procediments en conjunt, donat que de la mateixa manera que succeïa amb els procediments d'alumnes i professors, es tracta de scripts que realitzen funcions idèntiques tot i que interactuen amb taules diferents. En cadascun dels tres fitxers hi ha els tres corresponents scripts d'ABM.

Aquests són els procediments més senzills d'implementar. Quan es desitja donar d'**alta** un curs, una assignatura o una fita al calendari escolar, s'haurà d'introduir els atributs corresponents, els quals es poden visualitzar a l'esquema lògic de la figura 10.

La **baixa** d'un registre al calendari escolar és trivial, només es necessita especificar correctament els codis d'assignatura, curs i data per eliminar-ho del sistema. En el cas de donar de baixa una assignatura o curs, s'ha de tenir en compte el següent aspecte:

Hi ha moltes taules que referencien a la taula curs i a la taula assignatura segons l'esquema lògic de la figura 10. Davant d'aquesta situació, hi havia tres opcions possibles per implementar les baixes.

La primera opció era que quan es donés de baixa un curs o una assignatura es fes un DELETE en cascada (CASCADE). Aquesta opció es va menysprear donat que possibilita que es doni el cas de que hi hagi amonestacions que puguin tenir un curs o una assignatura *NULL*. I també matricules d'alumnes amb cursos *NULL*.

La segona opció és que el procediment d'eliminar un curs (o assignatura), consulti i elimini els registres de les taules on està referenciat (taula amonestació, taula imparteix, taula matricula, etc). És a dir, s'elimina el curs i a més s'eliminen totes les amonestacions, matricules, etc d'aquell curs. Aquesta opció seria com un DELETE en CASCADE però en comptes de posar NULLs a les claus foranes de les altres taules, el procediment visita totes les taules que apunten a *curs* i elimina els registres referenciats. Donat que es tracta d'una opció molt "radical", va ser descartada.

Finalment es va implementar l'opció més simple. Aquesta es basa en fer un simple DELETE del curs o assignatura entrat com a paràmetre del procediment. Es deixa la responsabilitat a l'usuari que utilitza la BD, de manera que no podrà eliminar un curs si aquest es troba referenciat en altres taules. Serà la responsabilitat de l'usuari eliminar prèviament les tuples de les altres taules que referencien al curs que es vol esborrar. És a dir, si s'intenta eliminar un curs però hi ha matricules vigents per aquell curs, el procediment fallarà. (violació de restricció d'integritat degut a la clau forana)

Per concloure el funcionament d'aquests arxius, indicar que la **modificació** en els tres casos és molt senzilla. Es sol·licita el codi o atributs que formen la clau primària per identificar inequívocament la tupla a modificar i les noves dades actualitzades. En el cas que les noves dades provoquin una excepció, es mostrarà per pantalla l'error.

3.2.5 Procediments d'ABM de les tipologies d'amonestacions i sancions

Tot i que els requeriments funcionals del sistema no expliciten específicament la implementació dels procediments per crear noves tipologies d'amonestacions i sancions, s'ha decidit desenvolupar-los.

Els procediments referents a les tipologies d'amonestacions són molt senzills i per aquest motiu només indicar que mitjançant el procediment d'**alta** del fitxer "*ABM tipus amonestacio.sql*" es defineix un nou tipus d'amonestació. Els procediments de **baixa** i **modificació** segueixen el mateix funcionament i estructura que els explicats en els apartats anteriors.

La taula *TipusAmonestacio* permet definir nous tipus d'amonestacions fàcilment, sense necessitat de canviar l'estructura de les taules o modificar els procediments emmagatzemats definits.

La taula *TipusAmonestacio* presentarà un aspecte similar a aquest:

CODI	TIPUS	DESCRIPCIO
1	ALUMNE_ARRIBA_TARD	L'ALUMNE HA ARRIBAT TARD A CLASSE
2	ALUMNE_SOROLL	L'ALUMNE HA FET MOLT DE SOROLL EN CLASSE
3	ALUMNE_MALPARLAT	L'ALUMNE HA SIGUT MALPARLAT
4	ALUMNE_NO_DEURES	L'ALUMNE NO HA FET ELS DEURES
5	ALUMNE_PORTA_MOBIL	L'ALUMNE HA PORTAT EL MÒBIL I L'UTILITZA DURANT UNA CLASSE
6	ALUMNE_VIDEOCONSOLA	L'ALUMNE HA PORTAT LA VIDEOCONSOLA A CLASSE

Figura 19

Els procediments referents a les tipologies de sancions són idèntics als anteriors i només es comentarà el aspecte més transcendental. Abans però, es mostrarà el contingut de la taula *TipusSancio*:

CODI	TIPUS	DESCRIPCIO	ACTIVACIO	TIPUSAMONESTACIO
1	HORA_EXTRA_ESTUDI	QUEDAR-SE UNA HORA EXTRA D'ESTUDI DURANT UNA SETMANA	3	1
2	PROVA_EXTRA	FER UNA PROVA D'AVUACIÓ EXTRA	5	2
3	TREBALL_EXTRA	FER UN TREBALL EXTRA PER ENTREGAR A FINAL DE CURS	7	3
4	TUTORIA_EXTRA	QUEDAR-SE CADA SETMANA DUES HORES AMB EL TUTOR	2	4
5	NO_TELEFON	ENTREGAR EL TELÈFON AL TUTOR FINS A FINAL DE CURS	2	5
6	DEURES_EXTRA	FER EXERCICIS EXTRA A FINAL DE SETMANA	6	1

Figura 20

Els atributs *activacio* i *tipusamonestacio* que es mostren a la imatge són fonamentals, donat que permeten emmagatzemar a la BD la definició de les regles per activar sancions automàtiques.

Per exemple el registre amb codi=1 de la taula anterior, indica que cada tres amonestacions que hi hagi del tipus 1 (*ALUMNE_ARRIBA_TARD*), s'apliqui el tipus de sanció *HORA_EXTRA_ESTUDI*.

3.2.6 Procediments d'ABM d'amonestacions

Els arxius “*ABM amonestacions.sql*” i “*Triggers Gestio Sancions Automatiques.sql*” constitueixen el nucli del sistema per gestionar les amonestacions i pel llançament de les sancions de manera automàtica.

La taula *Amonestacio* disposa de moltes claus foranes que referencien altres taules. Per aquest motiu l'**alta** d'una amonestació conté molts atributs formats per codis. Tots els atributs necessaris per fer una inserció es poden visualitzar a l'esquema lògic de la figura 10.

Un aspecte important és que tot i que la base de dades ha sigut dissenyada per suportar el règim d'amonestacions que defineixi cadascun dels centres educatius, el procediment d'alta no fa cap control sobre aquesta situació. Es podria haver implementar fàcilment aquesta funcionalitat, donat que l'únic que s'hauria de realitzar és fer una *SELECT* de la taula *AplicaAmonestacio* i a continuació obtenir el centre de l'amonestat amb un altre *SELECT* sobre la taula *Vincul*. Un cop fet aquestes accions, aleshores es comparen ambdós valors i si coincideixen vol dir que el centre suporta aquest tipus d'amonestació. El mateix raonament és vàlid per suportar el règim de sancions.

Altres limitacions del procediment d'alta són:

- No comprovació de que l'amonestat i l'amonestador pertanyin al mateix centre.
- No comprovació de que l'amonestat i l'amonestador pertanyin al mateix curs.
- No comprovació de que l'amonestat i l'amonestador pertanyin a la mateixa assignatura.

La no implementació d'aquestes comprovacions ve motivada per la gran despesa de temps que suposaria fer una inserció de dades que s'ajustés a totes aquestes restriccions. És a dir, en un entorn real serien fonamentals implementar-les, donat que en cas contrari la base de dades trencaria la seva consistència. Però per aquest projecte en concret, no s'ha estimat necessari “afilar tant prim”.

Arribat a aquest punt, s'explicarà com s'ha elaborat el mecanisme de sancions automàtiques. El codi encarregat d'aplicar les sancions automàtiques està format per dos dels quatre disparadors que es troben en el fitxer “*Triggers Gestio Sancions Automatiques.sql*”.

Mitjançant un disparador de tipus *BEFORE INSERT* es compta (immediatament abans d'inserir a la taula *Amonestacio* la nova amonestació) el número d'amonestacions que l'alumne ja tenia del mateix tipus que el de l'amonestació que li acaben d'imposar. El valor es desa en una variable global que servirà per comunicar dades amb altres disparadors.

Un cop s'ha realitzat l'operació anterior, s'executa un disparador de tipus *AFTER INSERT*. Fonamentalment la tasca que realitza aquest trigger és comprovar si l'amonestació imposada té gravetat *ALTA* o *NORMAL*. Si es dóna el primer cas, es procedeix a fer un *INSERT* a la taula *Sancio*, generant d'aquesta manera una sanció automàtica.

Si l'amonestació té gravetat *NORMAL*, aleshores el disparador utilitza la funció mòdul d'aquesta forma: *MOD* (el número de cops que l'alumne ha sigut amonestat pel mateix tipus d'amonestació / el número màxim de vegades a partir del qual l'amonestació genera una sanció).

Si el resultat del mòdul és zero, aleshores és que l'estudiant ha acumulat el nombre màxim d'amonestacions per un mateix tipus. Aleshores es procedeix a fer un INSERT a la taula Sancio, generant d'aquesta manera una sanció automàtica. En cas contrari no insereix la sanció.

Respecte el procediment de **baixa** d'amonestació, el únic fet destacable és que si l'amonestació que es desitja eliminar havia generat la creació d'una sanció, aleshores la sanció també es dona de baixa automàticament. Aquest mecanisme s'ha implementat gracies a la clau forana *codisancio* de la taula *Amonestacio*, la qual permet saber aquesta situació comprovant si l'atribut val NULL o no.

La **modificació** d'una amonestació s'ha implementat de manera que hi hagi unes certes limitacions per canviar alguns dels seus atributs. Concretament les limitacions d'aquest procediment són:

- No permet que una amonestació pugui canviar de gravetat
- No permet que una amonestació pugui passar d'un alumne a un altre diferent.
- No permet que una amonestació pugui passar d'un professor a un altre diferent.
- No permet que una amonestació pugui passar d'un curs a un altre diferent.
- No permet que una amonestació pugui passar d'una assignatura a una altre de diferent.
- No permet que una amonestació pugui canviar de tipus d'amonestació.

Si es desitja fer alguna d'aquestes operacions, aleshores l'amonestació haurà de ser esborrada del sistema i tornar a ser donada d'alta amb les dades desitjades.

La implementació d'aquestes comprovacions venen motivades per garantir la integritat de la base de dades i dels estadístics. Per exemple, canviar una amonestació de gravetat ALTA a NORMAL suposaria que la funció mòdul descrita en el procediment d'alta d'amonestació no realitzés la seva tasca de manera correcta.

Totes les restriccions anteriors són realitzades mitjançant un trigger de tipus *BEFORE UPDATE* sobre la taula *Amonestacio*. Sinó es compleixen aleshores es farà ROLLBACK de la transacció i es mostrarà el corresponent error.

3.2.7 Procediments d'ABM de sancions

El procediment d'**alta** de sanció no s'ha implementat donat que segons l'esquema relacional de la base de dades, no té sentit donar d'alta una sanció independent. És a dir, no pot existir una sanció si prèviament no s'ha donat d'alta l'amonestació que l'origina.

Quan es dona d'alta una amonestació amb gravetat ALTA o si es dona d'alta una amonestació NORMAL per a un alumne reincident, automàticament ja es dona d'alta la sanció.

Si es desitja donar d'alta una sanció, s'ha d'utilitzar el procediment *alta_amonestacio* del fitxer "ABM amonestacions.sql".

El mateix raonament segueix el procediment de **baixa** de sanció. Segons l'esquema relacional de la base de dades, no té sentit donar de baixa una sanció independent si prèviament no s'elimina l'amonestació que havia generat la sanció.

Quan es dona de baixa una amonestació amb gravetat ALTA o si es dona de baixa una amonestació NORMAL que havia generat una sanció per acumulació d'amonestacions, automàticament ja es dona de baixa la sanció vinculada a l'amonestació que es vol eliminar.

Si es desitja donar de baixa una sanció, s'ha d'utilitzar el procediment *baixa_amonestacio* del fitxer "*ABM amonestacions.sql*".

Finalment, la **modificació** d'una sanció és l'únic procediment que s'ha implementat. Permet canviar els valors dels registres de la taula *Sancio*. Aquest procediment però, té una restricció: El procediment evita que una sanció pugui canviar de tipus de sanció i el motiu és per garantir la integritat de la base de dades.

3.2.8 Gestió del Mòdul Estadístic

L'arxiu "*Gestio Modul Estadistic.sql*" crea vuit procediments encarregats d'actualitzar la informació estadística per quan es doni d'alta una amonestació o sanció. Existeix un procediment per cada estadístic sol·licitat als requeriments funcionals. A més també crea vuit procediments addicionals per realitzar l'operació contrària, és a dir, per actualitzar la informació estadística per quan es doni de baixa una amonestació o sanció.

Així doncs, el mòdul estadístic s'actualitza cada cop que una amonestació o sanció és creada o eliminada del sistema. És a dir, que les crides a les funcions per mantenir els estadístics actualitzats es realitzen només des dels procediments *alta_amonestacio* i *baixa_amonestacio*.

En el cas del procediment d'alta, les crides realitzades són:

```
-- Es procedeix a inserir l'amonestació
INSERT INTO Amonestacio VALUES ('',ins_datahora, i_comunicat_pares, i_gravetat, ins_codiamonestat,

-- S'actualitza l'estadístic E1
alta_e1(ins_codiamonestat, rsp2);
-- S'actualitza l'estadístic E3
alta_e3(ins_codiamonestador, ins_codicurs, rsp2);
-- S'actualitza l'estadístic E6
alta_e6(ins_codiamonestador, ins_codicurs, rsp2);
-- S'actualitza l'estadístic E8
alta_e8(rsp2);

-- Si s'ha generat una sanció, aleshores es crea la clau forana que serveix per vincular la sanció
IF (common_pkg.var4_global = 1) THEN
    UPDATE Amonestacio SET codisancio = common_pkg.var3_global WHERE CODI=common_pkg.var2_global;

-- S'actualitza l'estadístic E2
alta_e2(ins_codiamonestat, ins_codicurs, rsp2);
-- S'actualitza l'estadístic E4
alta_e4(ins_codicurs, rsp2);
-- S'actualitza l'estadístic E5
alta_e5(ins_codicurs, rsp2);
-- S'actualitza l'estadístic E7
alta_e7(ins_codicurs, rsp2);
END IF;
```

Figura 21

En el cas del procediment de baixa d'amonestació, les crides realitzades són:

```

-- Es procedeix a esborrar la tupla a la taula Amonestacio
DELETE FROM Amonestacio
WHERE (CODI=ins_codi_amonestacio);

-- S'actualitza l'estadístic E1
baixa_e1(i_codiamonestat, rsp2);
-- S'actualitza l'estadístic E3
baixa_e3(i_codiamonestador, i_codicurs, rsp2);
-- S'actualitza l'estadístic E6
baixa_e6(i_codiamonestador, i_codicurs, rsp2);
-- S'actualitza l'estadístic E8
baixa_e8(rsp2);

-- Si existeix una sanció vinculada a l'amonestació, aleshores també s'esborra
IF (i_comptar != 0) THEN

    DELETE FROM Sancio
    WHERE (CODI=i_sancio);

    -- S'actualitza l'estadístic E2
    baixa_e2(i_codiamonestat, i_codicurs, rsp2);
    -- S'actualitza l'estadístic E4
    baixa_e4(i_codicurs, rsp2);
    -- S'actualitza l'estadístic E5
    baixa_e5(i_codicurs, rsp2);
    -- S'actualitza l'estadístic E7
    baixa_e7(i_codicurs, rsp2);

```

Figura 22

Si es desitja saber com s'ha implementat cadascun dels procediments del mòdul estadístic, es pot veure als extensos comentaris del fitxer "*Gestio Modul Estadistic.sql*".

3.2.9 Procediments de consulta

El fitxer "*Procediments de consulta.sql*" conté els quatre procediments que donen resposta a les quatre consultes sol·licitades.

Els procediments de consulta seran comentats de manera molt breu, donat que no tenen cap misteri. Es basen en fer *SELECTS* de les diverses taules, segons la informació que es vulgui extreure.

Els quatre procediments tenen en comú la utilització de cursors explícits, donat que les consultes que realitzen poden retornar varis registres.

Els procediments al ser executats mostren per pantalla el resultat de la consulta. D'aquesta manera es facilita l'ús d'aquests als usuaris que necessitin consultar aquesta informació a la BD.

A continuació es mostra un exemple de resultat retornat per una d'aquestes consultes.

```
-- Llistat de tots els Alumnes d'un curs indicant-ne la seva informació bàsica.
DECLARE
    rsp varchar2(300);
BEGIN
    llistar_alumnes('5',rsp);
END;
```

anonymous block completed

```
-----
LLISTAT DE TOTS ELS ALUMNES DEL CURS AMB CODI=5
-----
```

CODI	DOCUMENTID	NACIONALITAT	COLLECTIU	NOM	NUM.EXPEDIENT
34	32658484N	ESPANYOLA	ALUMNE	MIREIA NEGRO ROSON	34
35	55687401N	ESPANYOLA	ALUMNE	MATIAS RIPOLL QUESO	35
36	67440487X	ESPANYOLA	ALUMNE	POL BARTRA BASETS	36
37	63087455G	ESPANYOLA	ALUMNE	JOSEP DALMASES CASTELLO	37
38	63178421Z	ESPANYOLA	ALUMNE	MARTA ALEMANY GUIRAL	38
39	66178887G	ESPANYOLA	ALUMNE	MANUEL PUIGMARTI PRIM	39
40	66301847W	ESPANYOLA	ALUMNE	DAMIA PUGES MORILLO	40
41	44507778G	ESPANYOLA	ALUMNE	CRISTIAN MUÑOZ JUVE	41
42	33448014W	ESPANYOLA	ALUMNE	LORENA LOPEZ MADRID	42
43	12845154S	ESPANYOLA	ALUMNE	OSCAR CONDE NUVIOLA	43
44	23598745F	ESPANYOLA	ALUMNE	PAU CASTILLO PUJOL	44
45	66154884Z	ESPANYOLA	ALUMNE	SARA FERROL TOLEDO	45
46	75751473N	ESPANYOLA	ALUMNE	AGUSTI SERRA RICO	46
47	88568901L	ESPANYOLA	ALUMNE	HUGO RECHE ORTEGA	47
48	44714900G	ESPANYOLA	ALUMNE	NURIA LOPEZ CURIEL	48
49	33221688X	ESPANYOLA	ALUMNE	ALEJANDRO TREFO GARRIDO	49
50	47810687J	ESPANYOLA	ALUMNE	CARLOS GAMOA GONZALEZ	50
51	89845780A	ESPANYOLA	ALUMNE	JOAQUIM ORTIZ MIRANDA	51
52	08066823Z	ESPANYOLA	ALUMNE	CARLA SURRIBAT SANS	52
53	85045667Z	ESPANYOLA	ALUMNE	FERNANDO BRITO CUENCA	53
54	87457812Q	ESPANYOLA	ALUMNE	MARIA DEL MAR GANDIA RUEDA	54
55	89542478S	ESPANYOLA	ALUMNE	JOSE BONACHO CASTAÑO	55
56	89124623S	ESPANYOLA	ALUMNE	SANTIAGO ZAMORANO PATO	56
57	78844087D	ESPANYOLA	ALUMNE	LEIRE PASILLO CASTRO	57
58	78874589A	ESPANYOLA	ALUMNE	JÀIME CINE HERRERAS	58
59	87365200T	ESPANYOLA	ALUMNE	DANIEL GARCIA GARCIA	59

OK

Figura 23

3.3 Joc de proves i errors

La fase d'implementació va culminar amb la realització de l'arxiu "Joc de proves i errors.sql" el qual conté diverses sentències independents PL/SQL per realitzar proves i crear situacions d'excepció.

Aquestes sentències es troben comentades en dit arxiu, i es troben dividides segons l'objectiu a exercir.

El joc de proves i errors va permetre confirmar que tant el disseny com la implementació realitzada complia amb tots els requeriments sol·licitats pel client. Després de la seva creació, es va donar per finalitzat el producte.

4. Recursos necessaris i valoració econòmica

4.1 Recursos necessaris

A continuació es detallaran cadascun dels recursos necessaris per a la realització del projecte. Per facilitar la posterior valoració econòmica d'aquests, hem classificat els diferents recursos en tres categories diferenciades: Recursos materials, tècnics i humans.

4.1.1 Recursos materials

Els recursos materials són els mitjans físics necessaris per l'assoliment dels objectius. En el nostre cas, l'objectiu és fer una proposta de disseny i implementació d'una base de dades.

Bàsicament només necessitem dos tipus de recursos materials: El maquinari i el programari.

El hardware o maquinari que serà utilitzat al llarg del projecte consta de les següents característiques tècniques:

- **Equip de sobretaula Acer Aspire T180-2B7H:** Processador AMD Athlon de 64 bits i doble nucli, amb una freqüència de 4200 MHz. Disc dur SATA de 250 GB. Memòria RAM de 2 GB DDR2.

En referència al software o programari que s'utilitzarà, a continuació especificuem el nom de cada programa i la seva funció:

- **Sistema Operatiu:** Microsoft Windows XP SP3
- **Sistema Gestor de Bases de Dades:** Oracle Express v10.2.0.1
- **Eina d'entorn gràfic de desenvolupament:** SQL Developer v1.0.0.15
- **Eina CASE de modelatge UML:** MagicDraw UML Personal Edition v16.0
- **Eina de gestió de projectes i elaboració de diagrames de Gantt:** GanttProject
- **Eina ofimàtica d'elaboració de documents:** Microsoft Office 2007 Professional
- **Eina de conversió de documents a format pdf:** PDF Creator
- **Eina de processament de text per escriure els scripts:** Notepad++

4.1.2 Recursos tècnics

Els recursos tècnics estan compostos per les tècniques i mètodes emprats per a la realització del treball.

Com hem comentat anteriorment, el model que seguirà aquest projecte és el conegut com a *Rational Unified Process*, el qual defineix una sèrie d'etapes clarament definides pel desenvolupament correcte del software i l'ordre en que es tenen que aplicar.

Dins del marc definit per aquest model, concretarem la metodologia de programació que aplicarem i especificarem l'ordre amb el que es duran a terme les diferents tècniques.

La metodologia emprada en la fase de construcció del software d'aquest projecte serà:

- 1- **Anàlisi i disseny:** Es defineixen les necessitats d'informació que haurà de cobrir el programari (què ha de fer el software) i es determina un disseny mitjançant l'algorítmica i les estructures de dades conegudes que doni resposta a aquestes necessitats. (com ho ha de fer).
- 2- **Programació:** Es procedeix a la codificació del disseny obtingut a l'etapa anterior mitjançant el llenguatge de programació PL/SQL
- 3- **Prova:** S'intenten recrear totes les possibles combinacions de condicions que es podrien donar en l'ús real del programari.
- 4- **Reprogramació:** Aquesta fase només serà necessària en el cas que es detectin anomalies o errors durant la fase de prova.
- 5- **Documentació:** Es documenta el tros de codi o script que s'acaba d'elaborar, fent anotacions en el mateix arxiu que conté el codi font.

Com a mitjans de suport a aquestes tècniques utilitzarem diagrames de flux, diagrames de classe i l'extensa documentació d'Oracle.

4.1.3 Recursos humans

Els recursos humans estan constituïts per totes les persones necessàries per a la realització del projecte. A més també aplega altres factors com el coneixement, l'experiència, habilitats, motivació, etc. de cadascun dels individus.

En aquest projecte l'equip humà estarà format només per una única persona (l'estudiant) que exercirà varis rols. Concretament haurà de fer els següents papers:

- **Cap de projecte:** En aquest treball el cap de projectes s'encarregarà de la planificació de les tasques, del compliment dels lliuraments acordats en les diferents dates, de la comunicació amb el client i de la coordinació amb la resta de rols.
- **Analista:** Les seves principals tasques seran la d'analitzar els requeriments que ha de complir el programari i crear els dissenys que donin una solució informàtica a les necessitats d'informació.
- **Programador:** Haurà de codificar els dissenys o algorismes que li proporcioni l'analista.
- **Tester:** S'encarregarà de la qualitat del software, de la preparació dels jocs de proves i de la correcció d'errors (bugs).
- **Documentalista:** La seva tasca principal serà l'elaboració de la memòria que recollirà tota la informació del projecte.
- **Comercial:** Serà l'encarregat d'elaborar la presentació del projecte.

D'altra banda l'entitat client estarà constituïda pel consultor del TFC, que realitzarà el paper de ser un membre del tribunal tècnic de la Generalitat de Catalunya. Posteriorment una sèrie d'experts en informàtica constituïran la resta del tribunal que avaluarà la proposta realitzada.

4.2 Valoració econòmica

El cost econòmic que suposarà aquest projecte ve determinat pels recursos materials i humans destinats a l'elaboració del mateix.

Els recursos materials estan formats pel hardware i el software. Pel que fa al maquinari només necessitem un equip, el qual té un cost aproximat d'uns 500 euros. En canvi el cost que suposa el software és menyspreable, donat que tot el programari i llicències necessàries són proporcionats per la UOC o bé són gratuïtament descarregables d'Internet.

Per calcular el cost destinat als recursos humans, haurem de disgregar el temps total estimat per la realització del projecte i repartir-ho entre els diferents rols. Recordem que l'estimació de temps era d'unes 270 hores en total.

A continuació es mostra com s'ha determinat el repartiment del temps total entre els diversos rols i el cost salarial associat al rol:

ROL	SALARI HORA	NOMBRE D'HORES	COST TOTAL
Cap de projecte	20	25	500
Analista	17	50	850
Programador	15	85	1275
Tester	12	30	360
Documentalista	10	60	600
Comercial	10	20	200
		270	3785

Com podem observar, el cost destinat als recursos humans serà aproximadament de 3785 euros.

Així doncs, si sumem a aquest valor el cost de 500 euros destinats als recursos materials, obtenim que la valoració econòmica o cost total del projecte serà de 4285 euros.

5. Conclusions

El primer que m'agradaria indicar és que un cop finalitzat el projecte, la sensació és de completa satisfacció personal.

L'elaboració d'aquest projecte m'ha donat l'oportunitat de sintetitzar tots els coneixements que tenia sobre el món de les bases de dades i que mai havien sigut utilitzats conjuntament per realitzar un projecte d'aquesta envergadura.

Considero que la majoria d'objectius definits a l'inici del TFC s'han assolit amb èxit, tot i que malauradament no s'ha pogut implementar millores per manca de temps.

Crec que podria haver gaudit més del treball si hagués tingut una mica més de temps lliure. M'hagués agradat millorar i optimitzar totes les funcionalitats, però això no vol dir pas que el treball entregat al client no sigui de molt bona qualitat.

Remarcant l'aspecte temporal, la planificació del projecte realitzada a la fase inicial de creació d'un pla de treball, ha sigut essencial per complir amb els terminis. Ho trobo una eina fonamental per no perdre l'orientació temporal del projecte al llarg del temps, i ajuda moltíssim per sapiguer si el ritme de feina que s'està portant és l'adequat o no.

Aquest projecte també m'ha ajudat a confirmar el meu interès especial per les bases de dades relacionals d'Oracle. Sobre aquest aspecte, continuaré formant-me per adquirir més coneixements d'aquest fabulós SGBD.

A nivell tècnic, el projecte s'ha pogut anar implementant sense gaires dificultats. El depurat disseny inicial de l'esquema conceptual ha sigut clau per posteriorment poder anar implementant totes les funcionalitats sense tenir que fer grans rectificacions ni perdre el temps amb canvis.

D'altra banda el tracte amb el client/consultor (Ismael Pérez Laguna) ha sigut excel·lent. Durant tots aquests mesos la comunicació ha sigut fluida i directe, fet que ha ajudat a que el producte desenvolupat sigui l'esperat per ambdós parts.

En definitiva, tot i que l'experiència ha comportat alguns moments d'angoixa per complir amb els terminis establerts, sincerament ha valgut la pena haver triat aquest TFC.

6. Glossari

- **BD (Base de Dades):** És un conjunt de dades que pertanyen a un mateix àmbit i que són emmagatzemades sistemàticament de manera persistent per a posteriors usos.
- **Model Entitat-Interrelació (E/R):** Model conceptual de dades. És una eina per al modelat de dades d'un sistema d'informació que emprà l'ús de diagrames estandarditzats.
- **Model Relacional:** Model de bases de dades que estructura les dades en relacions, tuples i atributs.
- **PL/SQL (Procedural Language/Structured Query Language):** Llenguatge de programació propietari d'Oracle que aporta noves característiques i enriqueix la programació amb SQL. Algunes de les funcionalitats més rellevants que inclou són l'ús de variables, la possibilitat de crear estructures modulars i de realitzar un control d'excepcions.
- **Script:** Arxiu de text que conté un conjunt d'instruccions d'un determinat llenguatge de programació.
- **SGBD (Sistema de Gestió de Base de Dades):** Programari complex que s'encarrega de proporcionar totes les eines necessàries per la creació, gestió i control d'una base de dades. Hi ha diversos SGBD en el mercat, cadascun dels quals sol ser d'un fabricant diferent. També hi ha alguns que són de programari lliure.
- **SQL (Structured Query Language / Llenguatge de consulta estructurat):** És un llenguatge declaratiu d'accés a bases de dades relacionals que permet especificar diversos tipus d'operacions sobre aquestes. Una de les seves característiques és l'ús de l'àlgebra i del càlcul relacional que permeten fer consultes per tal de recuperar o manipular d'una manera senzilla la informació d'interès d'una BD.
- **Stored Procedure (Procediment Emmagatzemat):** És un programa o conjunt d'instruccions el qual s'emmagatzema físicament en una base de dades. L'avantatge d'un Stored Procedure és que s'emmagatzema i s'executa directament en el SGBD, el qual usualment es troba funcionant en un servidor independent dels equips clients que l'utilitzen i juntament amb les dades. Al ser executat a petició d'un usuari o procés, només necessita enviar els resultats que genera de tornada a aquest, evitant la sobrecàrrega que comportaria la comunicació de grans quantitats de dades o instruccions d'entrada i sortida.
- **Trigger (Disparador):** És un procediment o conjunt d'instruccions que s'executa quan es compleix una condició establerta al realitzar una operació d'inserció (INSERT), actualització (UPDATE) o esborrat (DELETE). Algunes bases de dades poden executar triggers en crear, esborrar o editar usuaris, taules, bases de dades o altres objectes.
- **UML (Unified Modeling Language):** Llenguatge de modelatge de sistemes de programari. És un llenguatge gràfic per visualitzar, especificar, construir i documentar un sistema. És el més conegut i està en l'actualitat i ha sigut pres com a estàndard per la majoria de desenvolupadors. Una part d'aquest llenguatge està dedicada a la modelització conceptual de les dades.

7. Bibliografia

Materials UOC:

- Materials didàctics proporcionats a les assignatures: *Enginyeria del Programari*, *Bases de dades I*, *Bases de dades II* i *Sistemes de Gestió de Bases de Dades*.
- CD anomenat *Documentació Oracle v10.2.0.1* (amb codi CP06/11031/02009) proporcionat a l'assignatura *Sistemes de Gestió de Bases de Dades*.
- Document amb indicacions per realitzar un treball final de carrera:
http://materials.cv.uoc.edu/continguts/XW08_19018_00443/index.html
- Treball Final de Carrera anomenat *Diseño e Implementación de un Sistema de Comercio electrónico para una empresa farmacéutica*, de l'estudiant de la UOC *Eric Bergez* d'Enginyeria Tècnica en Informàtica de Gestió.
- Treball Final de Carrera anomenat *Disseny i Implementació d'un Sistema de Gestió de Competicions Esportives*, de l'estudiant de la UOC *Sergi Buyolo Freitas* d'Enginyeria Tècnica en Informàtica de Gestió.
- Treball Final de Carrera anomenat *Disseny i Implementació d'un Sistema de Gestió de Competicions Esportives*, de l'estudiant de la UOC *Ramon Gort Heredia* d'Enginyeria Tècnica en Informàtica de Gestió.

Fonts d'Internet amb informació genèrica sobre les BD:

- Informació sobre el model Rational Unified Process:
http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational
- Informació sobre el model entitat-interrelació:
http://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n
- Informació sobre el PL/SQL:
<http://es.wikipedia.org/wiki/PL/SQL>
<http://www.plsql.biz/>
- Informació sobre l'UML:
http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado
- Informació sobre el SQL:
<http://en.wikipedia.org/wiki/SQL>

Fonts d'Internet amb informació específica:

A continuació s'exposen varis enllaços emprats en la fase de programació i que contenen diversa informació sobre programar amb sentències PL/SQL d'Oracle:

- Utilització de la funció `SQLCODE` i el tractament d'excepcions:

http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14261/sqlcode_function.htm#CJAJEJGB
http://download.oracle.com/docs/cd/B19306_01/appdev.102/b14261/errors.htm#CJAJBAJG

- Com crear Stored Procedures:

http://www.sqlinfo.net/oracle/oracle_stored_procedure_INSERT.php
<http://www.techonthenet.com/oracle/procedures.php>

- Creació i utilització de Triggers:

http://www.oracle-base.com/articles/11g/TriggerEnhancements_11gR1.php
http://download.oracle.com/docs/cd/E12151_01/doc.150/e12155/triggers_proc_mysql.htm
http://psoug.org/reference/table_trigger.html
http://www.techonthenet.com/oracle/triggers/after_update.php

- Com emprar cursors explícits:

<http://www.devioker.com/contenidos/Tutorial-PLSQL/39/Cursores-Explicitos-en-PLSQL.aspx>

- Informació sobre l'ús de funcions:

<http://psoug.org/reference/functions.html>
http://psoug.org/reference/number_func.html
http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/functions088.htm

- Informació sobre l'ús de seqüències:

<http://publib.boulder.ibm.com/infocenter/iseries/v5r3/index.jsp?topic=%2Fdb2%2Frbafzmtscsequence.htm>
<http://www.techonthenet.com/oracle/sequences.php>
<http://publib.boulder.ibm.com/infocenter/idshelp/v10/index.jsp?topic=/com.ibm.sqls.doc/sqls270.htm>

- Com treballar amb camps de tipus data i de temps:

<http://infolab.stanford.edu/~ullman/fcdb/oracle/or-time.html>
http://download.oracle.com/docs/cd/B19306_01/server.102/b14225/ch4datetime.htm
<http://coba.dc.fi.udc.es/~penabad/bd1/node12.html>
http://psoug.org/reference/date_func.html
http://www.java2s.com/Tutorial/Oracle/0260_Date-Timestamp-Functions/TODATEwithINSERTstatement.htm

- Com resoldre determinats errors sorgits en la fase de programació:

http://mygnet.net/articulos/oracle/oracle_y_las_mutating_tables.757
<http://oracle.ittoolbox.com/documents/oracle-error-ora20000-13925>
<http://forums.oracle.com/forums/thread.jspa?threadID=687127>
<http://es.w3support.net/index.php?db=so&id=193107>

- Portals amb tot tipus d'informació i documentació sobre PL/SQL:

<http://www.orafaq.com/>
<http://www.oracle-base.com/>
http://download.oracle.com/docs/cd/B14117_01/nav/portal_2.htm