

**Disseny i implementació d'un sistema
de gestió de la base de dades de la
Colla Castellers Xiquets de Tarragona**

Autor: Oriol Sánchez Gil (ETIG)

Consultor: Jordi Ferrer Duran

Tarragona, 10 de juny de 2012

Força, equilibri, valor i senty

Resum

El present treball de final de carrera, en endavant TFC, tutelat dins de l'àrea de Bases de Dades Relacions, té com a objectiu el disseny i la implementació d'un sistema de gestió d'una base de dades que cobreixi les necessitats de la Colla Castellers Xiquets de Tarragona, en endavant Xiquets de Tarragona.

Aquestes necessitats son, en termes generals, que la base de dades pugui recollir d'una forma ordenada i eficient les dades del cens social i casteller així com tots aquells ens que siguin d'interès per als Xiquets de Tarragona. A la vegada, també es vol que aquesta informatització se li tregui un rendiment, sent capaç de realitzar les diferents gestions que pot tenir una entitat d'aquestes característiques.

A grans trets, la base de dades, haurà de ser capaç de resoldre diferents situacions, des de llistar tots els castellers actius, com a mostrar dades d'actuacions fins a emetre factures a entitats col·laboradores, entre d'altres.

Per aconseguir aquest objectiu se'ns demana el següent: dissenyar la base de dades, fer l'script de creació de les taules necessàries i implementar els procediments emmagatzemats que l'aplicació necessiti i utilitzi.

Índex de Continguts

Resum	3
1. Introducció	6
1.1. Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació	6
1.2. Objectius del TFC	7
1.2.1. <i>Objectius generals</i>	7
1.2.2. <i>Objectius específics</i>	7
1.3. Enfocament i mètode seguit	8
1.4. Planificació del projecte	9
1.4.1. <i>Dates importants</i>	9
1.4.2. <i>Identificació de tasques</i>	9
1.4.3. <i>Diagrama de Gantt</i>	11
1.5. Productes obtinguts	12
1.6. Breu descripció dels altres capítols de la memòria	12
2. Anàlisi de requisits	14
2.1. Descripció inicial	14
2.2. Requisits funcionals	14
2.2.1. <i>Entitat “Casteller”</i>	14
2.2.2. <i>Entitat “Actuació Castellera”</i>	15
2.2.3. <i>Entitat “Extern”</i>	16
3. Disseny de la Base de Dades	18
3.1. Disseny conceptual	18
3.1.1. <i>Identificació de les entitats</i>	18
3.1.2. <i>Diagrama UML</i>	18
3.1.3. <i>Atributs de les entitats</i>	20
3.1.4. <i>Restriccions d'integritat</i>	24
3.2. Disseny lògic	24
3.3. Disseny físic	26
3.3.1. <i>Creació de la tablespace</i>	26
3.3.2. <i>Creació de l'usuari</i>	26
3.3.3. <i>Creació de les taules</i>	27
3.3.4. <i>Creació dels índexs</i>	33

3.3.5. <i>Creació de les seqüències</i>	33
4. Implementació	34
4.1. Procediments taula Castell	34
4.2. Procediments taula Actuació	36
4.3. Procediments taula CastellsActuacio	38
4.4. Procediments taula Extern	41
4.5. Procediments taula Persona	43
4.6. Procediments taula Factura	45
4.7. Procediments taula Categoria	47
4.8. Procediments taula Casteller	49
4.9. Procediments taula CategoriaCasteller	51
4.10. Triggers	54
5. Joc de proves	56
6. Conclusions	63
7. Bibliografia	65

1. Introducció

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

Amb més de 40 anys d'història, els Xiquets de Tarragona, és la colla castellera més antiga de la ciutat de Tarragona, bressol de la tradició castellera.

En els últims anys, el fet casteller ha patit una transformació adreçada a assolir nous reptes, fent estructures cada cop més grans, que ha culminat en la proclamació com a Patrimoni Immaterial de la Humanitat per part de la UNESCO, l'any 2011.

Aquesta tendència de creixement, consistent en enfocar la feina a la consecució de les fites plantejades en cada temporada, fa palesa la necessitat d'establir mètodes i eines que facilitin el treball organitzatiu intern de les colles i que permetin fixar una metodologia sòlida a partir de la qual es puguin plantejar estratègies i nous reptes.

El fet que una colla castellera, per definició, sigui una entitat sense ànim de lucre, ha promogut una mancança de professionalització en els actors implicats que ara dificulta el treball de l'equip directiu de les colles.

El control de les diferents activitats i dels membres actius de l'entitat no està exclòs d'aquesta situació, doncs en el cas concret de la Colla Castellers Xiquets de Tarragona, la colla compta amb una base de dades rudimentària de la que se'n pot desprendre informació molt bàsica, amb una utilitat merament testimonial i moltes vegades desfasada.

Personalment, fa dos anys que sóc membre d'aquesta colla i sempre havia cregut convenient que el treball final de carrera tingués una utilitat real. Davant de la situació exposada, he cregut convenient i idoni aportar els meus coneixements informàtics i així poder col·laborar en la professionalització d'aquesta entitat, creant una projecte que comenci amb aquest treball i continuï a partir de la finalització d'aquest.

1.2. Objectius del TFC

1.2.1. Objectius generals

L'objectiu serà crear una base de dades que doni resposta a les necessitats actuals de l'entitat dels Xiquets de Tarragona, permetent conèixer no només la quantitat de membres de la colla, sinó que a la vegada sigui capaç de realitzar operacions amb les diferents dades emmagatzemades.

Per assolir els objectius, d'acord amb les premisses i tasques establertes inicialment, es realitzarà el següent:

- Disseny i implementació de la base de dades, realitzant els scripts de creació de les taules, índex, disparadors, etcètera. Creació dels procediments emmagatzemats necessaris per al bon funcionament de la base de dades.
- Implementació d'un mecanisme d'inicialització de la base de dades, creació d'un joc de proves exhaustiu que garanteixi el bon funcionament de les implementacions i control d'errors i excepcions.

Durant el procés de realització del TFC, es realitzaran diferents entregues, marcades dins del calendari esmentat en el punt 4 d'aquest apartat:

- **PAC1:** pla de treball: planificació real i detallada del projecte que es dura a terme durant els pròxims casi quatre mesos.
- **PAC2:** fase d'anàlisis i disseny
- **PAC3:** fase d'implementació i joc de proves

A més d'aquestes tres entregues, es lliurarà:

- **Memòria:** document de síntesis final
- **Presentació:** resum clar i concís del treball realitzat i dels resultats obtinguts durant la realització de la Memòria

1.2.2. Objectius específics

La realització d'aquest projecte, posarem en pràctica conceptes i coneixements adquirits en les assignatures de bases de dades (I, II i SGBD) i de desenvolupament de programari (Enginyeria del Programari, Tècniques de Desenvolupament de Programari i Informàtica Aplicada a la Gestió) amb el que s'intentarà assolir amb èxit el TFC i, en conseqüència, la finalització dels estudis d'Enginyeria Tècnica Informàtica de Gestió.

1.3. Enfocament i mètode seguit

Degut a les característiques d'aquest TFC, la metodologia que es durà a terme és la del cicle de vida en cascada.

Aquest mètode ens ha d'ajudar a adequar el projecte a una realitat el més aproximada possible, fixant terminis i equilibrant el volum de feina de cada fase.

Així doncs, tal i com ens mostra aquest mètode, el projecte constarà de les següents fases:

- Anàlisi de requisits
- Especificació i disseny conceptual
- Implementació i disseny lògic
- Joc de proves
- Manteniment i millora

Un cop destriades les fases del nostre projecte, s'escolliran les diferents tècniques que es duran a terme durant l'execució d'aquestes. Les tècniques respondran als coneixements adquirits en les diferents assignatures que tenen un efecte directe a l'àrea de Base de Dades Relacionals.

- Anàlisi de requisits mitjançant les tècniques utilitzades en l'assignatura d'Enginyeria del Programari i Informàtica Aplicada a la Gestió
- Disseny i implementació de bases de dades relacionals (disseny conceptual, diagrames Entitat/Relació, disseny lògic i de taules, normalització, etcètera) amb els diferents mètodes aplegats a les matèries de Bases de Dades I, II, SGBD, Tècniques de Desenvolupament de Programari així com a les diferents assignatures de programació.

1.4. Planificació del projecte

1.4.1. Dates importants

La planificació d'aquest projecte va acord amb l'avaluació continua proposada per la UOC i les dates que aquesta marca.

Data	Tasca
29/02/2012	Inici del projecte (realització de la PAC1)
18/03/2012	Entrega PAC1 (pla de treball)
19/03/2012	Inici PAC2
15/04/2012	Entrega PAC2
16/04/2012	Inici PAC3
20/05/2012	Entrega PAC3
21/05/2012	Inici Entrega Final
10/06/2012	Entrega Final

1.4.2. Identificació de tasques

Tenint en conte que el temps de realització del projecte es de gaire bé quatre mesos, s'estima que la dedicació setmanal serà d'unes 10 – 12 hores.

La planificació de tasques esta dividida en quatre parts:

PAC1

- Realització del pla de treball

PAC2:

- Anàlisis dels requisits
- Disseny de la BB.DD.
 - Disseny conceptual
 - Disseny lògic

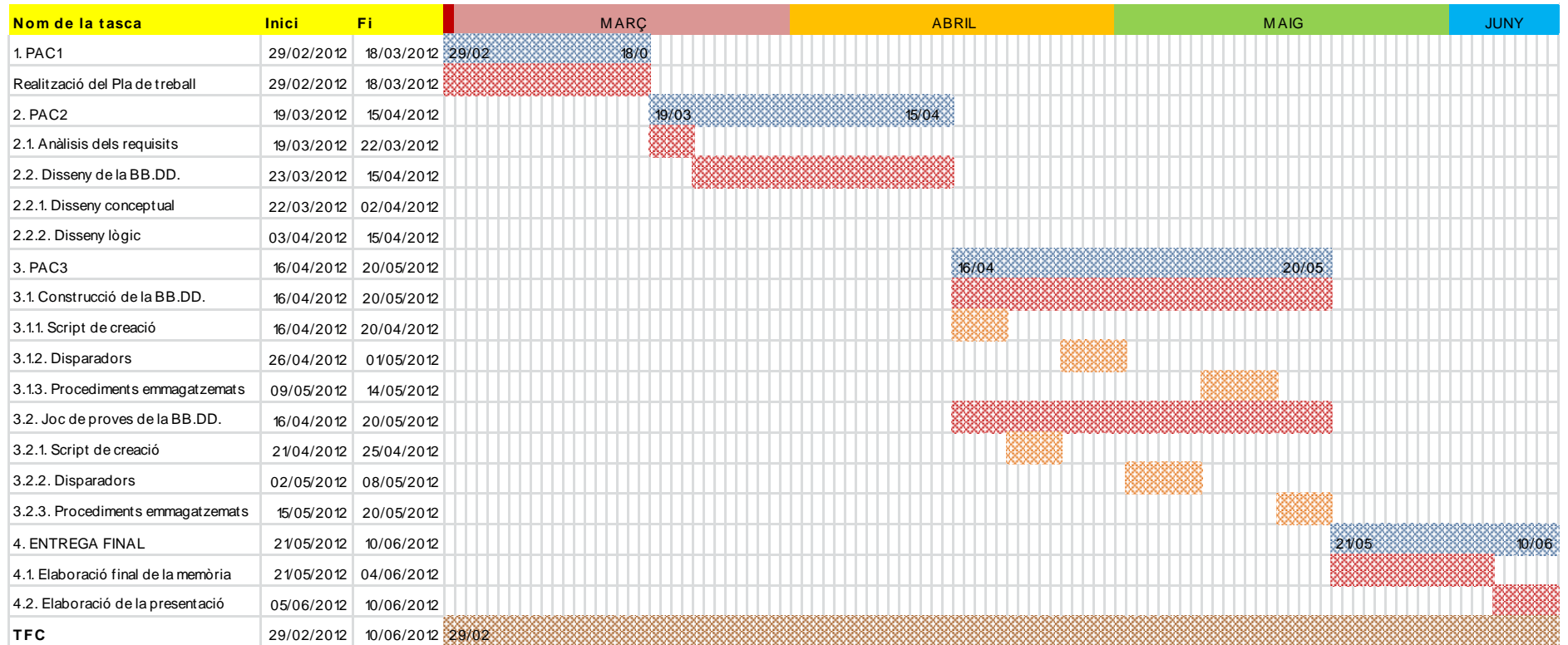
PAC3

- Construcció BB.DD.
 - Disseny físic
 - Scripts de creació
 - Disparadors
 - Procediments emmagatzemats
- Joc de Proves BB.DD.
 - Scripts de creació
 - Disparadors
 - Procediments emmagatzemats

ENTREGA FINAL

- Elaboració final de la memòria
- Elaboració de la presentació

1.4.3. Diagrama de Gantt



(*) Aquest diagrama es pot patir modificacions/ampliacions, a mesura que es desenvolupi el projecte.

1.5. Productes obtinguts

Els productes que s'obtidran a partir de l'execució d'aquest pla de treball, seran els següents:

- **Pla de treball.** Definició, planificació i estimació de las tasques que es duran a terme en aquest projecte.
- **Producte.** Base de Dades dels Xiquets de Tarragona
- **Memòria.** Recull estructurat de totes les activitats que s'han dut a terme durant el transcurs de la realització del projecte
- **Presentació.** Resum, en diapositives, dels resultats obtinguts

1.6. Breu descripció dels altres capítols de la memòria

Els possibles apartats futurs d'aquesta memòria son:

- Anàlisis dels requisits: reunions amb el client (Xiquets de Tarragona) serviran per esbrinar quins son els requeriments. En aquest apartat es farà la descripció d'aquests.
- Disseny de la base de dades: a partir dels requisits extrets del capítol anterior, es dissenyarà de forma conceptual i lògica el que serà la base de dades. En aquest apartat doncs, es mostrarà aquest disseny.
- Implementació: creació física del disseny conceptual a la base de dades. Implementació de les diferents funcionalitats que el sistema requereix (disparadors i procediments emmagatzemats) generant el control d'integritat i el control d'excepcions per al bon funcionament de la base de dades.
- Pla de contingència: anàlisis dels riscos durant i després de la realització del projecte i definició de les accions a dur a terme en cas d'error de la base de dades.
- Joc de proves: un cop estigui implementada la base de dades i les seves funcionalitats, es comprovarà el bon funcionament d'aquets, realitzant aquelles proves que posin en dubte el control d'integritat i les excepcions de la BB.DD.
- Valoració econòmica i recursos necessaris: exposició de la valoració del cost del producte i dels recursos necessaris per a realitzar amb èxit el projecte.

- Conclusions i continuïtat del projecte: conclusions personals i anàlisis de les possibilitats de la posada en marxa del projecte i del seu manteniment així com de les possibles millores o actualitzacions.

2. Anàlisi de requisits

2.1. Descripció inicial

Per tal de dur a terme l'anàlisi de requisits, partirem de tres grans "entitats" (Casteller, Actuació castellera i Extern) que considerem que son les susceptibles de ser tractades pel SGBD.

Aquest anàlisi es basarà en determinar en un primer moment les necessitats funcionals que s'extrauran de les entitats citades, i, posteriorment, es realitzarà el disseny (què emmagatzemarem) en funció d'aquestes funcionalitats.

2.2. Requisits funcionals

2.2.1. Entitat "Casteller"

Entenem l'entitat "Casteller" com aquella persona que desenvolupa el paper de casteller dins la colla. Per tant, aquesta entitat i totes les que se'n derivin, així com les funcionalitats que hi estiguin involucrades, aniran encarades a cobrir les necessitats de relació entre la colla i el casteller.

Per aquest motiu, les funcionalitats que han de garantir aquesta relació seran:

1. Manteniment de les dades (on es tindrà en compte):

- 1.1. Alta d'un nou casteller
- 1.2. Modificació d'un casteller
- 1.3. Baixa d'un casteller
- 1.4. Categorització del casteller per períodes
 - 1.4.1. Alta d'una nova categoria
 - 1.4.2. Modificació d'una categoria
 - 1.4.3. Baixa d'una categoria

2. Cerques multi-criteri:
 - 2.1. Cerques per casteller
 - 2.2. Cerques tècniques
 - 2.3. Cerques per categoria
 - 2.4. Altres cerques
3. Llistes (resultats de les cerques, ordenades per determinats criteris)
 - 3.1. Llistes de tots els castellers i les seves dades administratives
 - 3.2. Llistes de tots els castellers i les seves dades tècniques
 - 3.3. Llistes dels castellers d'una categoria determinada
 - 3.4. Llistes dels rebuts d'un casteller
 - 3.5. Altres llistes
4. Generar llistat (utilitzarem per defecte el format de Microsoft Excel)
5. Generar fitxa del casteller (utilitzarem per defecte el format de Microsoft Word)
6. Enviament de correus electrònics
 - 6.1. Enviament massiu (a tots els castellers que tenen correu electrònic)
 - 6.2. Enviament particular (enviament massiu amb parts del missatge diferents)
7. Enviament d'un missatge de mòbil
 - 7.1. Enviament massiu (a tots els castellers que tenen telèfon mòbil)
 - 7.2. Enviament particular (enviament massiu amb parts del missatge diferents)

2.2.2. Entitat "Actuació Castellera"

Les definició que podríem extreure de l'entitat "Actuació Castellera" seria: data on la colla realitza la seva activitat degut a un motiu concret.

Entendrem com activitat tot allò que faci estrictament referència a la construcció de castells o, altrament dit, torres humanes. Per tant, les funcionalitats que es citen a continuació van acord a la necessitat d'emmagatzemar les dades de les diferents diades i, posteriorment, treure'n un ús estadístic o tècnic.

1. Manteniment de les dades (on es tindrà en compte):
 - 1.1. Alta d'una nova actuació castellera
 - 1.2. Modificació d'una actuació castellera
 - 1.3. Baixa d'una diada castellera
 - 1.4. Castells d'una actuació castellera
 - 1.4.1. Alta d'un nou castell
 - 1.4.2. Modificació d'un castell
 - 1.4.3. Baixa d'un castell
2. Cerques multi-criteri:
 - 2.1. Cerques per actuació castellera
 - 2.2. Cerques per població
 - 2.3. Cerques per data i/o temporada
 - 2.4. Cerques per castell
 - 2.5. Altres cerques
3. Llistes (resultats de les cerques, ordenades per determinats criteris)
 - 3.1. Llistat de totes les actuacions d'una temporada
 - 3.2. Llistat de totes les actuacions entre dues dates
 - 3.3. Llistat de totes les actuacions a una població
 - 3.4. Llistat dels castells d'una temporada
 - 3.5. Llistat dels castells entre dues dates
 - 3.6. Altres llistes
4. Generar llistat (utilitzarem per defecte el format de Microsoft Excel)
5. Generar fitxa d'una diada (utilitzarem per defecte el format de Microsoft Word)

2.2.3. Entitat “Extern”

Anomenem a l'entitat “Extern” com aquelles empreses, institucions o colles castelleres on el client i manté una relació, a vegades no directament relacionada amb l'activitat castellera, però si necessària per la realització de la mateixa.

Aquesta entitat i les que se'n derivin, aniran directament relacionades a la consulta de dades i a la comptabilitat de la colla.

1. Manteniment de les dades (on es tindrà en compte):
 - 1.1. Alta d'un nou extern
 - 1.2. Modificació d'un extern
 - 1.3. Baixa d'un extern
 - 1.4. Facturació a un extern
 - 1.4.1. Alta d'una nova factura
 - 1.4.2. Modificació d'una factura
 - 1.4.3. Baixa d'una factura
2. Cerques multi-criteri
 - 2.1. Cerca per extern
 - 2.2. Cerca per data i/o temporada
 - 2.3. Cerca per factura
 - 2.4. Cerca per tipus d'extern
 - 2.5. Altres cerques
3. Llistes (resultats de les cerques, ordenades per determinats criteris)
 - 3.1. Llistes de les factures d'una temporada
 - 3.2. Llistes de les factures d'un extern
 - 3.3. Llistes de les factures d'un extern en una temporada
 - 3.4. Llistes de les factures d'un extern en dues dates determinades
 - 3.5. Llistes per tipus d'extern
 - 3.6. Altres llistes
4. Generar llistat (utilitzarem per defecte el format de Microsoft Excel)
5. Generar fitxa d'una empresa (utilitzarem per defecte el format de Microsoft Word)
6. Generar una factura a una empresa (utilitzarem per defecte el format de Microsoft Word)
7. Enviament de correus electrònics (nomes per colla castellera)
 - 7.1. Enviament massiu (a totes les colles castellers que tenen correu electrònic)
 - 7.2. Enviament particular (enviament massiu amb parts del missatge diferents)

3. Disseny de la Base de Dades

3.1. Disseny conceptual

3.1.1. Identificació de les entitats

Després de l'anàlisi anterior, identifiquem dos tipus d'entitats, les principals i les secundàries. Les principals són les més importants del sistema i les secundàries són les que complementen a les primeres.

Així doncs, identifiquem les següents entitats principals:

- Casteller
- Actuació
- Empresa

També tenim les entitats secundàries, que ens ajuden al funcionament de les primeres:

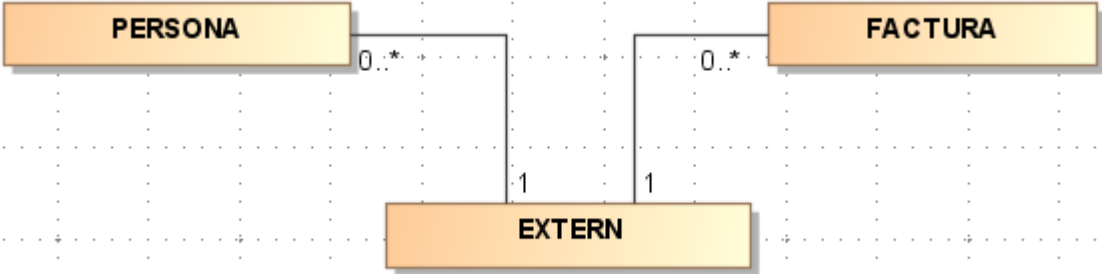
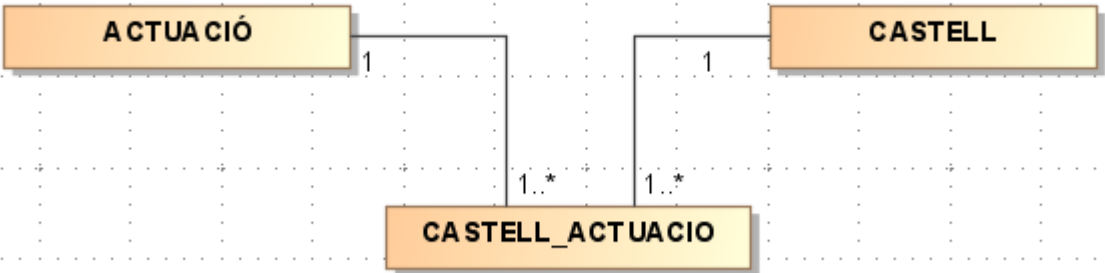
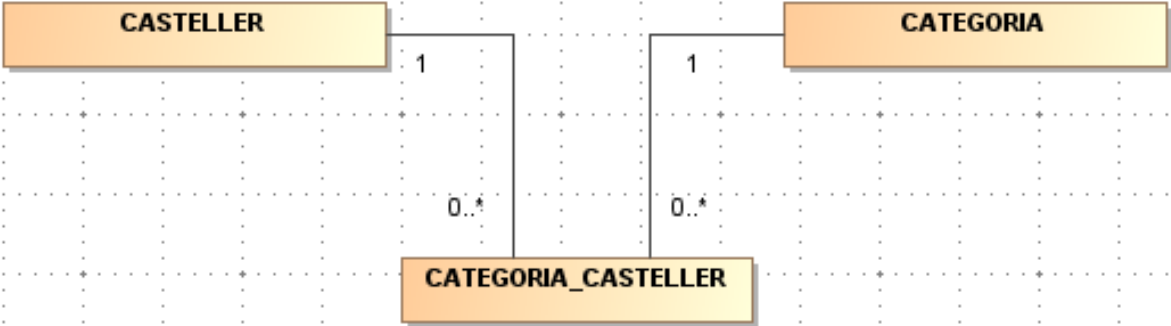
- Castell
- Categoria
- Persona
- Factura

I també les que existiran per la unió de dos entitats anteriorment esmentades:

- Castells Actuació
- Categories Casteller

3.1.2. Diagrama UML

En aquest diagrama plasmarem les explicacions anteriors:



3.1.3. Atributs de les entitats

Entitat	CASTELLER			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_casteller</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
dni	String	Document d'identitat	Null	Qualsevol cadena de caràcters
nom	String	Nom	Not Null	Qualsevol cadena de caràcters
cognoms	String	Cognoms	Not Null	Qualsevol cadena de caràcters
data_naixement	Date	Data de naixement	Not Null	Qualsevol data (format dd/mm/aaaa)
sexe	String	Sexe	Not Null	Home/Dona
alies	String	Sobrenom del casteller	Null	Qualsevol cadena de caràcters
alçada	Integer	Alçada en cm	Null	Qualsevol enter
pes	Integer	Pes en kg	Null	Qualsevol enter
adreça	String	Adreça	Null	Qualsevol cadena de caràcters
codi_postal	String	Codi Postal	Null	Qualsevol cadena de caràcters
poblacio	String	Població	Null	Qualsevol cadena de caràcters
pais	String	País	Null	Qualsevol cadena de caràcters
telefon_1	Integer	Telèfon principal	Null	Qualsevol cadena d'enters
telefon_2	Integer	Telèfon secundari	Null	Qualsevol cadena d'enters
correu_electronic	String	Correu electrònic	Null	Qualsevol cadena de caràcters (amb "@")
data_alta	Date	Data d'alta del casteller	Not Null	Qualsevol data (format dd/mm/aaaa)
actiu	Booleà	Mostra si el casteller està actiu o no	Not Null	SI/NO
soci_casteller	Booleà	Mostra si el casteller és soci o no	Null	SI/NO
quota	Decimal	Mostra la quantita de quota	Null	Qualsevol cadena decimal (soci_casteller SI)
num_compte	Integer	Número de compte	Null	Qualsevol cadena d'enters
baixa	Booleà	Mostra si el casteller està actiu o no	Null	SI/NO
motiu_baixa	String	Descripció del motiu de la baixa	Null	Administrativa/Defunció/Altres (Baixa SI)
pinya_mans	Booleà		Null	SI/NO
pinya_dau	Booleà		Null	SI/NO
pinya_lateral	Booleà		Null	SI/NO
pinya_home_darrera	Booleà		Null	SI/NO
pinya_crossa	Booleà		Null	SI/NO
pinya_agulla	Booleà		Null	SI/NO
observacions	String	Observacions que es vulguin anotar	Null	Qualsevol cadena de caràcters

Entitat	CATEGORIA			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_categoria</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
nom	String	Nom de la categoria	Not Null	Qualsevol cadena de caràcters
descripcio	String	Descripció de la categoria	Null	Qualsevol cadena de caràcters

Entitat	CATEGORIA_CASTELLER			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_cat_casteller</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
data_inici	Date	Data inici d'una categoria d'un casteller	Not Null	Qualsevol data (format dd/mm/aaaa)
data_fi	Date	Data fi d'una categoria d'un casteller	Null	Qualsevol data (format dd/mm/aaaa)
observacions	String	Descripció de la categoria	Null	Qualsevol cadena de caràcters

Entitat	EXTERN			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_extern</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
CIF	String	Document d'identitat	Not Null	Qualsevol cadena de caràcters
nom	String	Nom	Not Null	Qualsevol cadena de caràcters
adreça	String	Adreça	Null	Qualsevol cadena de caràcters
codi_postal	String	Codi Postal	Null	Qualsevol cadena de caràcters
poblacio	String	Població	Null	Qualsevol cadena de caràcters
pais	String	País	Null	Qualsevol cadena de caràcters
telefon	Integer	Telèfon principal	Null	Qualsevol cadena d'enters
fax	Integer	Fax principal	Null	Qualsevol cadena d'enters
correu_electronic	String	Correu electronic	Null	Qualsevol cadena de caràcters (amb "@")
pagina_web	String	Pàgina web de l'extern	Null	Qualsevol cadena de caràcters
num_compte	Integer	Número de compte	Null	Qualsevol cadena d'enters
observacions	String	Observacions que es vulguin anotar	Null	Qualsevol cadena de caràcters
tipus_extern	String	Tipus d'extern	Not Null	Colla Castellera/Institució/Client-Proveïdor

Entitat	PERSONA			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_persona</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
nom	String	Nom	Not Null	Qualsevol cadena de caràcters
cognoms	String	Cognoms	Not Null	Qualsevol cadena de caràcters
telefon_1	Integer	Telèfon secundari	Null	Qualsevol cadena d'enters
telefon_2	Integer	Telèfon secundari	Null	Qualsevol cadena d'enters
correu_electronic	String	Correu electronic	Null	Qualsevol cadena de caràcters (amb "@")
observacions	String	Observacions que es vulguin anotar	Null	Qualsevol cadena de caràcters

Entitat	FACTURA			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_factura</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
import	String	Nom	Not Null	Qualsevol cadena de caràcters
tipus_factura	String	Si es a pagar o a cobrar	Not Null	Pagar/Cobrar
referència	String	Referència de la factura	Not Null	Qualsevol cadena de caràcters
observacions	String	Observacions que es vulguin anotar	Null	Qualsevol cadena de caràcters

Entitat	ACTUACIO			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_actuacio</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
nom	String	Nom	Not Null	Qualsevol cadena de caràcters
data	Date	Data que es celebra l'actuació	Not Null	Qualsevol data (format dd/mm/aaaa)
temporada	Integer	Temporada de l'actuació	Not Null	Qualsevol cadena d'enters
motiu	String	Motiu pel qual es fa l'actuació	Not Null	Diada castellera/
població	String	Població on es fa l'actuació	Not Null	Qualsevol cadena de caràcters
observacions	String	Observacions que es vulguin anotar	Null	Qualsevol cadena de caràcters

Entitat	CASTELL			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_castell</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
nom	String	Nom	Not Null	Qualsevol cadena de caràcters
castellers_tronc	Integer	Número de castellers de tronc	Not Null	Qualsevol cadena d'enters
casteller_pinya	Integer	Número de castellers de pinya (mínim)	Not Null	Qualsevol cadena d'enters
punts carregat	Integer	Puntuació del castell carregat	Not Null	Qualsevol cadena d'enters
punts descarregat	String	Puntuació del castell descarregat	Not Null	Qualsevol cadena d'enters

Entitat	CASTELLS_ACTUACIO			
Atribut	Tipus	Descripció	Null/Not Null	Valors permesos
<i>id_cast_actuacio</i>	String	Identificador únic generat pel sistema	Not Null	Qualsevol cadena de caràcters (amb lògica)
estat	String	Descripció del castell	Not Null	Carregat/Descarregat/Intent
Puntuació	Integer	Puntuació del Castell	Not Null	Qualsevol cadena d'enters
observacions	String	Observacions que es vulguin anotar	Null	Qualsevol cadena de caràcters

3.1.4. Restriccions d'integritat

Durant la fase d'anàlisi de requisits s'ha fet una recopilació de les restriccions d' integritat que hauria de contemplar el disseny segons les característiques el client. Son les següents:

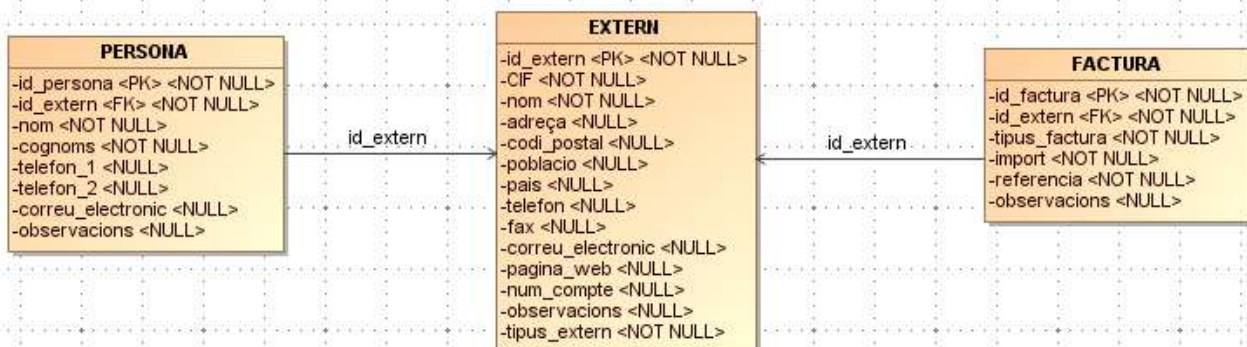
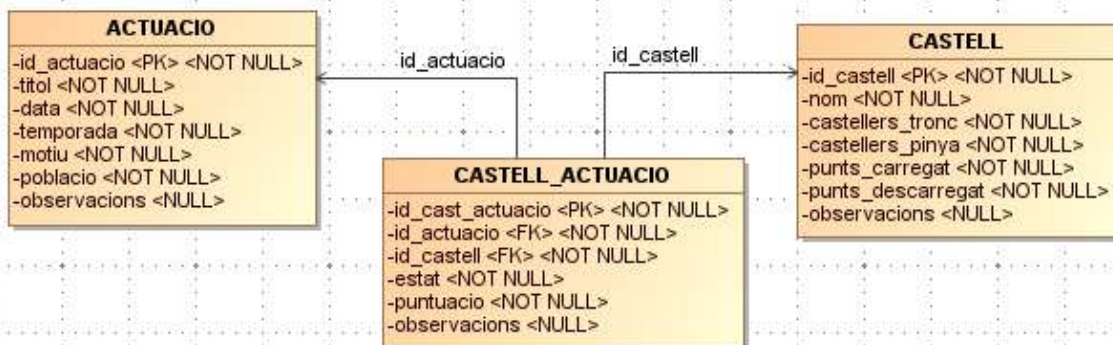
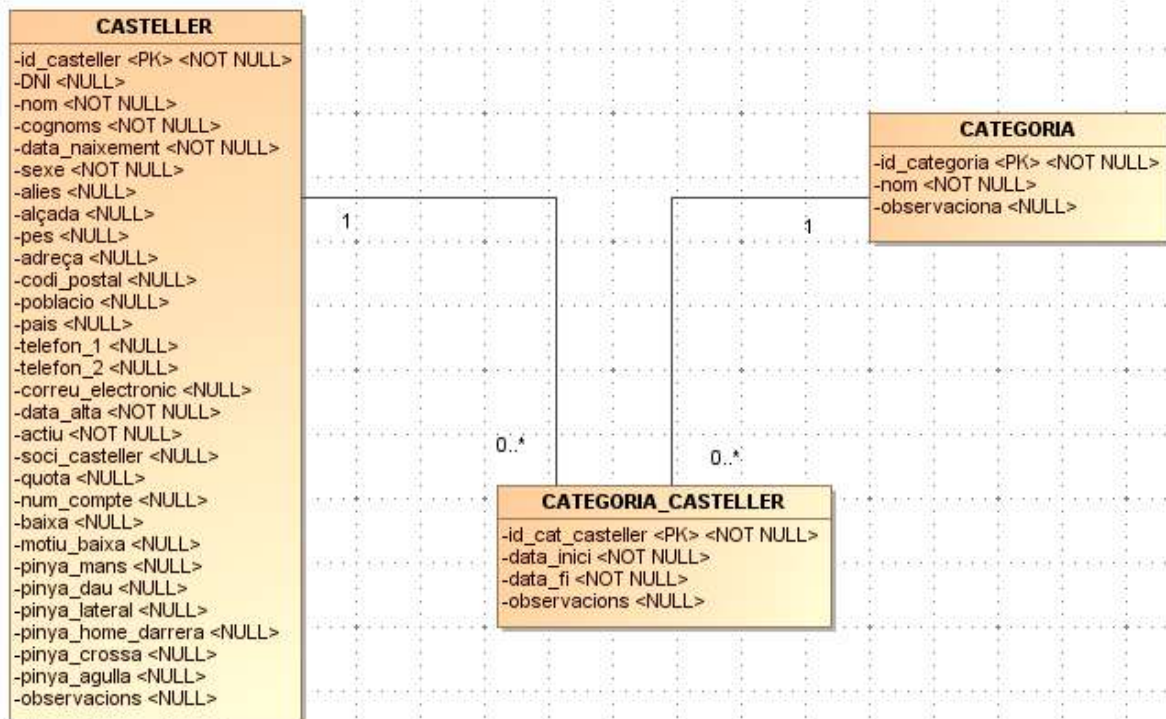
- A l'entitat categoria, la data inicial a de ser anterior de la data final
- A l'entitat casteller, solament es podrà indicar el valor de la quota, si l'atribut soci_casteller es SI
- A l'entitat casteller, solament es podrà indicar el motiu_baixa, si l'atribut baixa, està activat

A part, tant el l'apartat anterior, com en diagrama UML, s'han establert les restriccions d'integritat a través de la declaració de NOT NULL i de les multiplicitats de les interrelacions

3.2. Disseny lògic

El disseny lògic pren com a punt de partida el diagrama UML i la descripció de les entitats i atributs del disseny conceptual, per a transformar-lo en un conjunt de relacions i claus a partir d'unes regles bàsiques que s' apliquen sobre les entitats i les seves interrelacions. Les regles que seguirem són:

- Les entitats originen relacions
- Les interrelacions binaries 1:1 i 1:N originen claus foranes
- Les interrelacions binaries M:N originen noves relacions
- En les entitats dèbils, la clau forana forma part de la clau primària
- Les generalitzacions/especialitzacions generen una relació per a l'entitat superclasse i una relació per cadascuna de les subclasses
- Transformació de les entitats associatives en les transformacions de les interrelacions a les que depenen
- Nomenclatura de les restriccions: PK (primary key), FK (foreign key), NULL (nul), NOT NULL (no nul)
- Les multiplicitats i cardinalitats de les relacions es mantenen iguals però canvia la forma de presentació: s'indica afegint una fletxa.



3.3. Disseny físic

En aquest apartat, plasmarem el disseny lògic en taules i així crearem la nostra base de dades. Per tal de realitzar aquest procediment hem escollit el SGBD Oracle.

3.3.1. Creació de la tablespace

Es crearan dos 'TABLESPACE', un per les dades i un altre pels índexs, que actuaran d'unitats lògiques d'emmagatzematge dins de la base de dades que ens proporciona el mateix sistema ORACLE.

Les tablespace es crearan gestionats localment i amb una gestió automàtica d'emmagatzematge, degut al seu alt rendiment i la sencillez de l'assignació d'espai que ens permet. Aquest tindran una mida inicial de 10M, que s'estima suficient per a les dades inicialment introduïdes.

```
CREATE TABLESPACE XIQUETS_DAT DATAFILE 'C:\ORACLEXE\ORADATA\XE\XIQUETS_DAT_01.DBF' SIZE 10M AUTOEXTEND OFF
LOGGING
PERMANENT
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
BLOCKSIZE 8k
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;
```

```
CREATE TABLESPACE XIQUETS_IND DATAFILE 'C:\ORACLEXE\ORADATA\XE\XIQUETS_IND_01.DBF' SIZE 10M AUTOEXTEND OFF
LOGGING
PERMANENT
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
BLOCKSIZE 8k
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;
```

3.3.2. Creació de l'usuari

Inicialment, entenem aquesta BBDD per a un ús local i en un lloc 'estàtic' sense la necessitat que existeixin varis usuaris operant en el mateix moment. Per aquest motiu, considerem que creant un sol usuari complirem els requisits i les necessitats que el client ens planteja. Aquest usuari tindrà la capacitat de accedir i manipular les taules i les funcions de l'aplicació essent el

propietari de les mateixes. Per aquest motiu doncs, l'usuari haurà de gaudir dels privilegis necessaris d'accés a les tablespaces (creades anteriorment).

Crearem doncs, un usuari (XIQUETS) amb contrasenya (XIQUETS1970) amb els rols CONNECT i RESOURCE assignats i amb quota il·limitada sobre les tablespaces.

```
DROP USER (IF EXISTS) XIQUETS CASCADE;

CREATE USER XIQUETS
IDENTIFIED BY 'XIQUETS1970'
DEFAULT TABLESPACE XIQUETS_DAT
TEMPORARY TABLESPACE TEMP
PROFILE DEFAULT
ACCOUNT UNLOCK;

GRANT RESOURCE TO XIQUETS;
GRANT CONNECT TO XIQUETS;
GRANT CREATE VIEW TO XIQUETS;
ALTER USER XIQUETS DEFAULT ROLE ALL;

ALTER USER XIQUETS QUOTA UNLIMITED ON XIQUETS_DAT;
ALTER USER XIQUETS QUOTA UNLIMITED ON XIQUETS_IND;
```

3.3.3. Creació de les taules

Utilitzant les sentències que ens proporciona l'Oracle realitzarem el procés de la creació de les taules i les seves claus primàries i forànies.

Tindrem en compte les següents coses:

- El tipus i la mida de les taules es tractarà de forma particular en cada cas
- Es marcaran NOT NULL aquells camps que siguin imprescindibles i que no puguin quedar buits
- S'utilitzarà el flag CHECK en aquells camps on sigui necessari assegurar-se que el contingut s'ajusta als requisits.

A continuació es detallen les taules que es crearan a la base de dades. El script de creació d'aquestes, és l'anomenat "3.CreacioTaules.sql" que s'adjunta al document. Com podem veure, la realització del disseny físic ha generat una sèrie de canvis a l'estructura, en comparació a les entitats presentades dins del disseny lògic

```

-----
----- TAULA CASTELL -----
-----

DROP TABLE castell CASCADE CONSTRAINTS;

CREATE TABLE castell
(
idCastell NUMBER NOT NULL,
nomCastell VARCHAR2 (50) NOT NULL,
castTronc NUMBER,
castPinya NUMBER,
puntsCarregat NUMBER NOT NULL,
puntsDescarregat NUMBER NOT NULL,
baixaCastell VARCHAR2 (1)
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE castell
ADD CONSTRAINT PK_castell PRIMARY KEY (idCastell)
USING INDEX
TABLESPACE XIQUETS_IND;

ALTER TABLE castell
ADD CONSTRAINT CK_baixaCastell_castell CHECK (baixaCastell IN ('S'));

-----
----- TAULA ACTUACIO -----
-----

DROP TABLE actuacio CASCADE CONSTRAINTS;

CREATE TABLE actuacio
(
idActuacio NUMBER NOT NULL,
nomActuacio VARCHAR2 (50) NOT NULL,
dataActuacio DATE NOT NULL,
temporada NUMBER (4) NOT NULL,
tipusActuacio VARCHAR2 (25) NOT NULL,
tipusAmfitrio VARCHAR2 (25) NOT NULL,
nomAmfitrio VARCHAR2 (50),
poblacioActuacio VARCHAR2 (30) NOT NULL,
baixaActuacio VARCHAR2 (1)
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE actuacio
ADD CONSTRAINT PK_actuacio PRIMARY KEY (idActuacio)
USING INDEX
TABLESPACE XIQUETS_IND;

ALTER TABLE actuacio
ADD CONSTRAINT CK_tipusActuacio_actuacio CHECK (tipusActuacio IN ('Diada','Congres','Altres'));

ALTER TABLE actuacio
ADD CONSTRAINT CK_tipusAmfitrio_actuacio CHECK (tipusAmfitrio IN ('Xiquets','Colla
Castellera','Empresa','Entitat','Institucio'));

ALTER TABLE actuacio
ADD CONSTRAINT CK_baixaActuacio_actuacio CHECK (baixaActuacio IN ('S'));

-----
-- TAULA CASTELLSACTUACIO --
-----

DROP TABLE castellsActuacio CASCADE CONSTRAINTS;

CREATE TABLE castellsActuacio
(
idCastAct NUMBER NOT NULL,

```

```

estatCastAct VARCHAR2 (25) NOT NULL,
obsCastAct VARCHAR2 (150),
puntuacio NUMBER,
caActuacio NUMBER NOT NULL,
caCastell NUMBER NOT NULL
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE castellsActuacio
  ADD CONSTRAINT PK_castellsActuacio PRIMARY KEY (idCastAct)
  USING INDEX
  TABLESPACE XIQUETS_IND;

ALTER TABLE castellsActuacio
  ADD CONSTRAINT FK_castell_castellsActuacio FOREIGN KEY (caCastell)
  REFERENCES Castell (idCastell);

ALTER TABLE castellsActuacio
  ADD CONSTRAINT FK_actuacio_castellsActuacio FOREIGN KEY (caActuacio)
  REFERENCES Actuacio (idActuacio);

ALTER TABLE castellsActuacio
  ADD CONSTRAINT CK_estat_castellsActuacio CHECK (estatCastAct IN ('CARREGAT','DESCARREGAT','INTENT'));

-----
-----  TAULA EXTERN  -----
-----

DROP TABLE extern CASCADE CONSTRAINTS;

CREATE TABLE extern
(
idExtern NUMBER NOT NULL,
cif NUMBER NOT NULL,
nomExtern VARCHAR2 (50) NOT NULL,
tipusExtern VARCHAR2 (30) NOT NULL,
adrecaExtern VARCHAR2 (100),
cpExtern VARCHAR2 (10),
poblExtern VARCHAR2 (30),
paisExtern VARCHAR2 (20),
telefonExtern NUMBER (9),
faxExtern NUMBER (9),
webExtern VARCHAR2 (30),
ccExtern NUMBER (20),
obsExtern VARCHAR2 (150),
baixaExtern VARCHAR2 (1)
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE extern
  ADD CONSTRAINT PK_extern PRIMARY KEY (idExtern)
  USING INDEX
  TABLESPACE XIQUETS_IND;

ALTER TABLE extern
  ADD CONSTRAINT CK_tipusExtern_extern CHECK (tipusExtern IN ('Colla
Castellera','Empresa','Entitat','Institucio','Particular'));

ALTER TABLE extern
  ADD CONSTRAINT CK_baixaExtern_extern CHECK (baixaExtern IN ('S'));

-----
-----  TAULA PERSONA  -----
-----

DROP TABLE persona CASCADE CONSTRAINTS;

CREATE TABLE persona
(
idPersona NUMBER NOT NULL,

```

```

nomPersona VARCHAR2 (30) NOT NULL,
cognomsPersona VARCHAR2 (60) NOT NULL,
telf1Persona NUMBER (9),
telf2Persona NUMBER (9),
mailPersona VARCHAR2 (30),
pExtern NUMBER NOT NULL
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE persona
  ADD CONSTRAINT PK_persona PRIMARY KEY (idPersona)
  USING INDEX
  TABLESPACE XIQUETS_IND;

ALTER TABLE persona
  ADD CONSTRAINT FK_extern_persona FOREIGN KEY (pExtern)
  REFERENCES extern (idExtern);

-----
----- TAULA FACTURA -----
-----

DROP TABLE factura CASCADE CONSTRAINTS;

CREATE TABLE factura
(
idFactura NUMBER NOT NULL,
importFactura NUMBER NOT NULL,
tipusFactura VARCHAR2 (6) NOT NULL,
referencia NUMBER (20) NOT NULL,
fExtern NUMBER NOT NULL
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE factura
  ADD CONSTRAINT PK_factura PRIMARY KEY (idFactura)
  USING INDEX
  TABLESPACE XIQUETS_IND;

ALTER TABLE factura
  ADD CONSTRAINT FK_extern_factura FOREIGN KEY (fExtern)
  REFERENCES extern (idExtern);

ALTER TABLE factura
  ADD CONSTRAINT CK_tipusFactura_factura CHECK (tipusFactura IN ('COBRAR','PAGAR'));

-----
----- TAULA CATEGORIA -----
-----

DROP TABLE categoria CASCADE CONSTRAINTS;

CREATE TABLE categoria
(
idCategoria NUMBER NOT NULL,
nomCategoria VARCHAR2 (25) NOT NULL,
desCategoria VARCHAR2 (100),
baixaCat VARCHAR2 (1)
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE categoria
  ADD CONSTRAINT PK_categoria PRIMARY KEY (idCategoria)
  USING INDEX
  TABLESPACE XIQUETS_IND;

ALTER TABLE categoria
  ADD CONSTRAINT CK_baixacat_categoria CHECK (baixaCat IN ('S'));
    
```

```

-----
----- TAULA CASTELLER -----
-----

DROP TABLE casteller CASCADE CONSTRAINTS;

CREATE TABLE casteller
(
idCasteller NUMBER NOT NULL,
dni VARCHAR2 (15) NOT NULL,
nomCast VARCHAR2 (30) NOT NULL,
cognomsCast VARCHAR2 (60) NOT NULL ,
dNaixement DATE NOT NULL,
sexe VARCHAR2 (4) NOT NULL,
alies VARCHAR2 (20),
altura NUMBER (4),
pes NUMBER (4),
adrecaCast VARCHAR2 (100),
cpCast NUMBER (5),
poblCast VARCHAR2 (30),
paisCast VARCHAR2 (20),
telf1Cast NUMBER (9),
telf2Cast NUMBER (9),
mailCast VARCHAR2 (30),
dAlta DATE NOT NULL,
actiu VARCHAR2 (1) NOT NULL,
sociCasteller VARCHAR2 (1),
import NUMBER,
ccCast NUMBER (20),
baixaCast VARCHAR2 (1),
motiuBaixa VARCHAR2 (15),
mans VARCHAR2 (1),
dau VARCHAR2 (1),
lateral VARCHAR2 (1),
homeDarrera VARCHAR2 (1),
crossa VARCHAR2 (1),
agulla VARCHAR2 (1),
obsCasteller VARCHAR2 (100)
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE casteller
ADD CONSTRAINT PK_casteller PRIMARY KEY (idCasteller)
USING INDEX
TABLESPACE XIQUETS_IND;

ALTER TABLE casteller
ADD CONSTRAINT CK_sexe_casteller CHECK (sexe IN ('HOME','DONA'));

ALTER TABLE casteller
ADD CONSTRAINT CK_motiuBaixa_casteller CHECK (motiuBaixa IN ('DEFUNCIO','ADMINISTRATIVA','ALTRES'));

ALTER TABLE casteller
ADD CONSTRAINT CK_actiu_casteller CHECK (actiu IN ('S','N'));

ALTER TABLE casteller
ADD CONSTRAINT CK_sociCasteller_casteller CHECK (sociCasteller IN ('S','N'));

ALTER TABLE casteller
ADD CONSTRAINT CK_baixa_casteller CHECK (baixa IN ('S'));

ALTER TABLE casteller
ADD CONSTRAINT CK_mans_casteller CHECK (mans IN ('S','N'));

ALTER TABLE casteller
ADD CONSTRAINT CK_dau_casteller CHECK (dau IN ('S','N'));

ALTER TABLE casteller
ADD CONSTRAINT CK_lateral_casteller CHECK (lateral IN ('S','N'));

```

```

ALTER TABLE casteller
  ADD CONSTRAINT CK_homeDarrera_casteller CHECK (homeDarrera IN ('S','N'));

ALTER TABLE casteller
  ADD CONSTRAINT CK_crossa_casteller CHECK (crossa IN ('S','N'));

ALTER TABLE casteller
  ADD CONSTRAINT CK_agulla_casteller CHECK (agulla IN ('S','N'));

-----
--TAULA CATEGORIACASTELLER --
-----

DROP TABLE categoriaCasteller CASCADE CONSTRAINTS;

CREATE TABLE categoriaCasteller
(
idCatCast NUMBER NOT NULL,
dIniCatCast DATE NOT NULL,
dFiCatCast DATE,
obsCatCast VARCHAR2 (150),
ccCategoria NUMBER NOT NULL,
ccCasteller NUMBER NOT NULL
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE categoriaCasteller
  ADD CONSTRAINT PK_categoriaCasteller PRIMARY KEY (idCatCast)
  USING INDEX
  TABLESPACE XIQUETS_IND;

ALTER TABLE categoriaCasteller
  ADD CONSTRAINT FK_cat_categoriaCasteller FOREIGN KEY (ccCategoria)
  REFERENCES categoria (idCategoria);

ALTER TABLE categoriaCasteller
  ADD CONSTRAINT FK_cast_categoriaCasteller FOREIGN KEY (ccCasteller)
  REFERENCES casteller (idCasteller);

-----
----- TAULA REBUT -----
-----

DROP TABLE rebut CASCADE CONSTRAINTS;

CREATE TABLE rebut
(
idRebut NUMBER NOT NULL,
importRebut NUMBER NOT NULL,
tipusRebut VARCHAR2 (6) NOT NULL,
rCasteller NUMBER NOT NULL
)
TABLESPACE XIQUETS_DAT;

ALTER TABLE rebut
  ADD CONSTRAINT PK_rebut PRIMARY KEY (idRebut)
  USING INDEX
  TABLESPACE XIQUETS_IND;

ALTER TABLE rebut
  ADD CONSTRAINT FK_casteller_rebut FOREIGN KEY (rCasteller)
  REFERENCES casteller (idCasteller);

ALTER TABLE rebut
  ADD CONSTRAINT CK_tipusRebut_rebut CHECK (tipusRebut IN ('COBRAR','PAGAR'));

```


4. Implementació

Una vegada hem dissenyat les taules que tindrà el nostre sistema, en aquesta fase realitzarem el codi necessari perquè aquestes funcionin correctament.

Tot aquest codi està adjuntat en forma de scripts al present document, on es detalla el funcionament arxiu per arxiu.

Els procediments emmagatzemats tindran paràmetres d'entrada i solament un paràmetre de sortida. Aquest paràmetre serà un indicador de com ha anat el procediment i, si es el cas, per quin motiu ha fallat.

4.1. Procediments taula Castell

4.1.1. Alta Castell

Package: XIQUETS.PKG_ACTUACIO

Procedure: altaCastell

Propòsit: Dóna d'alta un nou castell si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID; 3. El valor dels punts de castell descarregat és superior al valor dels punts del castell carregat)

Paràmetres d'entrada: p_nomCastell (NOT NULL), p_castTronc, p_castPinya, p_puntsCarregat (NOT NULL), p_puntsDescarregat (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta un nou castell. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres], que no existeix un castell amb les mateixes dades [en aquest cas el nom] o que el valor dels punts del castell descarregat és superior al valor dels punts del castell carregat) i si tot es correcte s'afegeix un registre a la base de dades i el procediment envia un

missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_castellExistent: Ja existeix un registre amb diferent ID
3. E_castellPunts: Els punts descarregat no pot ser inferior o igual als punts carregat

4.1.2. Modifica Castell

Package: XIQUETS.PKG_ACTUACIO

Procedure: modifCastell

Propòsit: Modifica un castell si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. El valor dels punts de castell descarregat és superior al valor dels punts del castell carregat; 4. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idCastell (NOT NULL), p_nomCastell (NOT NULL), p_castTronc, p_castPinya, p_puntsCarregat (NOT NULL), p_puntsDescarregat (NOT NULL).

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar un castell. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador, que el valor dels punts del castell descarregat és superior al valor dels punts del castell carregat o que la modificació no comporti tenir un registre repetit [en aquest cas el nom]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_castellNoExistent: No existeix el registre que es passa per paràmetre
3. E_castellPunts: Els punts descarregat no pot ser inferior o igual als punts carregat

4. E_castellRepetit: No pot haver un mateix castell amb ID diferent

4.1.3. Baixa Castell

Package: XIQUETS.PKG_ACTUACIO

Procedure: baixaCastell

Propòsit: Dóna de baixa (eliminació lògica) un castell si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. El registre no s'havia donat de baixa amb anterioritat)

Paràmetres d'entrada: p_idCastell (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa un castell. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que el castell no estigui donat de baixa amb anterioritat) i si tot es correcte es modifica el registre (dada baixaCastell = 'S') i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_castellNoExistent: No existeix el registre que es passa per paràmetre
3. E_castellBaixa: El registre ja s'havia donat de baixa

4.2. Procediments taula Actuacio

4.2.1. Alta Actuació

Package: XIQUETS.PKG_ACTUACIO

Procedure: altaActuacio

Propòsit: Dóna d'alta una nova actuació si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID)

Paràmetres d'entrada: p_idActuacio (NOT NULL), p_nomActuacio (NOT NULL), p_dataActuacio (NOT NULL), p_temporada (NOT NULL), p_tipusActuacio (NOT NULL), p_tipusAmfitrio (NOT NULL), p_nomAmfitrio, p_poblacioActuacio (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta una nova actuació. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres] i que no existeix una actuació amb les mateixes dades [en aquest cas el nom, la data i la població]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_actuacioExistent: Ja existeix un registre amb diferent ID

4.2.2. Modifica Actuació

Package: XIQUETS.PKG_ACTUACIO

Procedure: modifActuacio

Propòsit: Modifica una actuació si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idActuacio (NOT NULL), p_nomActuacio (NOT NULL), p_dataActuacio (NOT NULL), p_temporada (NOT NULL), p_tipusActuacio (NOT NULL), p_tipusAmfitrio (NOT NULL), p_nomAmfitrio, p_poblacioActuacio (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar una actuació. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas el nom, la data i la població]) i si tot es

correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_actuacioNoExistent: No existeix el registre que es passa per paràmetre
3. E_actuacioRepetida: No pot haver una mateixa actuació amb ID diferent

4.2.3. Baixa Actuació

Package: XIQUETS.PKG_ACTUACIO

Procedure: baixaActuacio

Propòsit: Dóna de baixa (eliminació lògica) una actuació si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. El registre no s'havia donat de baixa amb anterioritat)

Paràmetres d'entrada: p_idActuacio (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa un castell. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que l'actuació no estigui donada de baixa amb anterioritat) i si tot es correcte es modifica el registre (dada baixaActuacio = 'S') i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_actuacioNoExistent: No existeix el registre que es passa per paràmetre
3. E_actuacioBaixa: El registre ja s'havia donat de baixa

4.3. Procediments taula CastellsActuacio

4.3.1. Alta CastellsActuacio

Package: XIQUETS.PKG_ACTUACIO

Procedure: altaCastAct

Propòsit: Dóna d'alta un castell a una actuació si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El castell que es passa per paràmetre existeix a la base de dades; 3. L'actuació que es passa per paràmetre existeix a la base de dades; 4. El registre no s'hagi inclòs prèviament amb un altre ID)

Paràmetres d'entrada: p_estatCastAct (NOT NULL), p_obsCastAct, p_caActuacio (NOT NULL), p_caCastell (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta un nou castell a una actuació. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres], que existeix el identificador del castell, que existeix el identificador de l'actuació i que no existeix un castell d'una actuació amb les mateixes dades[en aquest cas l'estat del castell]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_ca_actuacioNoExistent: No existeix l'actuació que es passa per paràmetre
3. E_ca_castellNoExistent : No existeix el castell que es passa per paràmetre
4. E_caExistent: Ja existeix un registre amb diferent ID

4.3.2. Modifica CastellsActuacio

Package: XIQUETS.PKG_ACTUACIO

Procedure: modifCastAct

Propòsit: Modifica un castell a una actuació si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID))

Paràmetres d'entrada: p_idCastAct (NOT NULL), p_estatCastAct (NOT NULL), p_obsCastAct

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar un castell d'una actuació. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas l'estat del castell]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_caNoExistent: No existeix el registre que es passa per paràmetre
3. E_caRepetit: No pot haver una mateix castell d'una actuació amb ID diferent

4.3.3. Baixa CastellsActuacio

Package: XIQUETS.PKG_ACTUACIO

Procedure: baixaCastAct

Propòsit: Dóna de baixa (eliminació física) un castell d'una actuació si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades)

Paràmetres d'entrada: p_idCastAct (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa un castell d'una actuació. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador] o que existeix el identificador) i

si tot es correcte es procedeix a realitzar la eliminació física del registre i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_caNoExistent: No existeix el registre que es passa per paràmetre

4.4. Procediments taula Extern

4.4.1. Alta Extern

Package: XIQUETS.PKG_EXTERN

Procedure: altaExtern

Propòsit: Dóna d'alta un nou extern si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID)

Paràmetres d'entrada: p_cif (NOT NULL), p_nomExtern (NOT NULL), p_tipusExtern (NOT NULL), p_adrecaExtern, p_cpExtern, p_poblExtern, p_paisExtern, p_telefonExtern, p_faxExtern, p_webExtern, p_ccExtern, p_obsExtern

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta un nou extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres] i que no existeix un extern amb les mateixes dades [en aquest cas el CIF]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_externExistent: Ja existeix un registre amb diferent ID

4.4.2. Modifica Extern

Package: XIQUETS.PKG_EXTERN

Procedure: modifExtern

Propòsit: Modifica un extern si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idExtern (NOT NULL), p_cif (NOT NULL), p_nomExtern (NOT NULL), p_tipusExtern (NOT NULL), p_adrecaExtern, p_cpExtern, p_poblExtern, p_paisExtern, p_telefonExtern, p_faxExtern, p_webExtern, p_ccExtern, p_obsExtern

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar un extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas el CIF]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_externNoExistent: No existeix el registre que es passa per paràmetre
3. E_externRepetit: No pot haver un mateix extern amb ID diferent

4.4.3. Baixa Extern

Package: XIQUETS.PKG_EXTERN

Procedure: baixaExtern

Propòsit: Dóna de baixa (eliminació lògica) un extern si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. El registre no s'havia donat de baixa amb anterioritat)

Paràmetres d'entrada: p_idExtern (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa un extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que l'extern no estigui donat de baixa amb anterioritat) i si tot es correcte es modifica el registre (dada baixaExtern = 'S') i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_externNoExistent: No existeix el registre que es passa per paràmetre
3. E_externBaixa: El registre ja s'havia donat de baixa

4.5. Procediments taula Persona

4.5.1. Alta Persona

Package: XIQUETS.PKG_EXTERN

Procedure: altaPersona

Propòsit: Dóna d'alta un nova persona si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID; 3. L'extern que es passa per paràmetre existeix a la base de dades)

Paràmetres d'entrada: p_nomPersona (NOT NULL), p_CognomsPersona (NOT NULL), p_telf1Persona, p_telf2Persona, p_mailPersona, p_pExtern (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta una persona a un extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres], que existeix el identificador de l'extern, i que no existeix una persona d'un extern amb les mateixes dades [en aquest cas nom i cognoms]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable

de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_personaExistent: Ja existeix un registre amb diferent ID
3. E_pExternNoExistent: No existeix l'extern que es passa per paràmetre

4.5.2. Modifica Persona

Package: XIQUETS.PKG_EXTERN

Procedure: modifPersona

Propòsit: Modifica una persona si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idPersona (NOT NULL), p_nomPersona (NOT NULL), p_CognomsPersona (NOT NULL), p_telf1Persona, p_telf2Persona, p_mailPersona

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar un persona d'un extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas el nom i cognoms]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_personaNoExistent: No existeix el registre que es passa per paràmetre
3. E_personaRepetida: No pot haver una mateixa persona amb ID diferent

4.5.3. Baixa Persona

Package: XIQUETS.PKG_EXTERN

Procedure: baixaPersona

Propòsit: Dóna de baixa (eliminació física) una persona si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades)

Paràmetres d'entrada: p_idPersona (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa una persona d'un extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador] o que existeix el identificador) i si tot es correcte es procedeix a realitzar la eliminació física del registre i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_personaNoExistent: No existeix el registre que es passa per paràmetre

4.6. Procediments taula Factura

4.6.1. Alta Factura

Package: XIQUETS.PKG_EXTERN

Procedure: altaFactura

Propòsit: Dóna d'alta un nova factura si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID; 3. L'extern que es passa per paràmetre existeix a la base de dades)

Paràmetres d'entrada: p_importFactura (NOT NULL), p_tipusFactura (NOT NULL), p_referencia (NOT NULL), p_fExtern (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta una factura a un extern. El procediment comprova

que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres], que existeix el identificador de l'extern, i que no existeix una factura d'un extern amb les mateixes dades [en aquest la referencia]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_facturaExistent: Ja existeix un registre amb diferent ID
3. E_fExternNoExistent: No existeix l'extern que es passa per paràmetre

4.6.2. Modifica Factura

Package: XIQUETS.PKG_EXTERN

Procedure: modifFactura

Propòsit: Modifica una factura si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idFactura (NOT NULL), p_importFactura (NOT NULL), p_tipusFactura (NOT NULL), p_referencia (NOT NULL),

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar una factura d'un extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas la referencia]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor

2. E_facturaNoExistent: No existeix el registre que es passa per paràmetre
3. E_facturaRepetida: No pot haver una mateixa factura amb ID diferent

4.6.3. Baixa Factura

Package: XIQUETS.PKG_EXTERN

Procedure: baixaFactura

Propòsit: Dóna de baixa (eliminació física) una factura si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades)

Paràmetres d'entrada: p_idFactura (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa una factura d'un extern. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador] o que existeix el identificador) i si tot es correcte es procedeix a realitzar la eliminació física del registre i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_facturaNoExistent: No existeix el registre que es passa per paràmetre

4.7. Procediments taula Categoria

4.7.1. Alta Categoria

Package: XIQUETS.PKG_CASTELLER

Procedure: altaCategoria

Propòsit: Dóna d'alta una nova categoria si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID)

Paràmetres d'entrada: p_nomCategoria (NOT NULL), p_desCategoria

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta una nova categoria. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres] i que no existeix una categoria amb les mateixes dades [en aquest cas el nom]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_categoriaExistent: Ja existeix un registre amb diferent ID

4.7.2. Modifica Categoria

Package: XIQUETS.PKG_CASTELLER

Procedure: modifCategoria

Propòsit: Modifica una categoria si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idCategoria (NOT NULL), p_nomCategoria (NOT NULL), p_desCategoria

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar una categoria. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas el nom]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor

2. E_categoriaNoExistent: No existeix el registre que es passa per paràmetre
3. E_categoriaRepetida: No pot haver una mateixa categoria amb ID diferent

4.7.3. Baixa Categoria

Package: XIQUETS.PKG_CASTELLER

Procedure: baixaCategoria

Propòsit: Dóna de baixa (eliminació lògica) una categoria si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. El registre no s'havia donat de baixa amb anterioritat)

Paràmetres d'entrada: p_idCategoria (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa una categoria. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la categoria no estigui donada de baixa amb anterioritat) i si tot es correcte es modifica el registre (dada baixaCategoria = 'S') i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_categoriaNoExistent: No existeix el registre que es passa per paràmetre
3. E_categoriaBaixa: El registre ja s'havia donat de baixa

4.8. Procediments taula Casteller

4.8.1. Alta Casteller

Package: XIQUETS.PKG_CASTELLER

Procedure: altaCasteller

Propòsit: Dóna d'alta un nou casteller si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID)

Paràmetres d'entrada: p_dni (NOT NULL), p_nomCast (NOT NULL), p_cognomsCast (NOT NULL). p_dNaixement (NOT NULL), p_sexe (NOT NULL), p_alies, p_altura, p_pes, p_adrecaCast, p_cpCast, p_poblCast, p_paisCast, p_telf1Cast, p_telf2Cast, p_mailCast, p_dAlta (NOT NULL), p_actiu (NOT NULL), p_sociCasteller, p_import, p_ccCast, p_mans, p_dau, p_lateral, p_homeDarrera, p_crossa, p_agulla, p_obsCasteller

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta un nou casteller. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres] i que no existeix un casteller amb les mateixes dades [en aquest cas el DNI]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_castellerExistent: Ja existeix un registre amb diferent ID

4.8.2. Modifica Casteller

Package: XIQUETS.PKG_CASTELLER

Procedure: modifCasteller

Propòsit: Modifica un casteller si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idCasteller (NOT NULL), p_dni (NOT NULL), p_nomCast (NOT NULL), p_cognomsCast (NOT NULL). p_dNaixement (NOT NULL), p_sexe (NOT NULL), p_alies, p_altura, p_pes, p_adrecaCast, p_cpCast, p_poblCast, p_paisCast, p_telf1Cast, p_telf2Cast, p_mailCast, p_dAlta (NOT NULL), p_actiu (NOT NULL), p_sociCasteller, p_import, p_ccCast, p_mans, p_dau, p_lateral, p_homeDarrera, p_crossa, p_agulla, p_obsCasteller

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar un casteller. El procediment comprova que es

compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas el DNI]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_castellerNoExistent: No existeix el registre que es passa per paràmetre
3. E_castellerRepetit: No pot haver un mateix casteller amb ID diferent

4.8.3. Baixa Casteller

Package: XIQUETS.PKG_CASTELLER

Procedure: baixaCasteller

Propòsit: Dóna de baixa (eliminació lògica) una categoria si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. El registre no s'havia donat de baixa amb anterioritat)

Paràmetres d'entrada: p_idCasteller (NOT NULL), p_motiuBaixa

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa d'un casteller. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que el casteller no estigui donat de baixa amb anterioritat) i si tot es correcte es modifica el registre (dada baixaCasteller = 'S') i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_castellerNoExistent: No existeix el registre que es passa per paràmetre
3. E_castellerBaixa: El registre ja s'havia donat de baixa

4.9. Procediments taula CategoriaCasteller

4.9.1. Alta CategoriaCasteller

Package: XIQUETS.PKG_CASTELLER

Procedure: altaCatCast

Propòsit: Dóna d'alta una nova categoria a un casteller si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre no s'hagi inclòs prèviament amb un altre ID; 3. La categoria que es passa per paràmetre existeix a la base de dades; 4. El casteller que es passa per paràmetre existeix a la base de dades)

Paràmetres d'entrada: p_dIniCatCast (NOT NULL), p_obsCatCast, p_ccCategoria (NOT NULL), p_ccCasteller (NOT NULL)

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar d'alta una nova categoria a un casteller. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL que no tingui valor [en aquest cas tots els paràmetres], que existeix el identificador de la categoria, que existeix el identificador del casteller i que no existeix una categoria d'un casteller amb les mateixes dades [en aquest cas la data d'inici de la categoria]) i si tot es correcte, s'afegeix un registre a la base de dades i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_ccExistent: Ja existeix un registre amb diferent ID
3. E_cc_categoriaNoExistent: No existeix la categoria que es passa per paràmetre
4. E_cc_castellerNoExistent: No existeix el casteller que es passa per paràmetre

4.9.2. Modifica CategoriaCasteller

Package: XIQUETS.PKG_CASTELLER

Procedure: modifCatCast

Propòsit: Modifica una categoria d'un casteller si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La modificació no implica tenir un registre repetit amb diferent ID)

Paràmetres d'entrada: p_idCatCast (NOT NULL), p_dIniCatCast (NOT NULL), p_obsCatCast

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui modificar una categoria d'un casteller. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador], que existeix el identificador o que la modificació no comporti tenir un registre repetit [en aquest cas la data d'inici de la categoria]) i si tot es correcte es modifiquen les dades del registre indicat i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_ccNoExistent: No existeix el registre que es passa per paràmetre
3. E_ccRepetit: No pot haver una mateixa categoria d'un casteller amb ID diferent

4.9.3. Baixa CategoriaCasteller

Package: XIQUETS.PKG_CASTELLER

Procedure: baixaCatCast

Propòsit: Dóna de baixa una categoria d'un casteller que, en aquest cas, significa posar la data de finalització. Això succeirà si prèviament ha superat uns requisits (1. Tots els camps obligatoris tenen valor; 2. El registre existeix a la base de dades; 3. La data fi es posterior a la data inicial)

Paràmetres d'entrada: p_idCatCast (NOT NULL), p_dFiCatCast

Paràmetre de sortida: RSP

Funcionament: La funció passa per paràmetre totes les dades mínimament necessàries (NOT NULL) per tal de que es pugui donar de baixa un categoria. El procediment comprova que es compleixin els requisits (que no hi hagi un camp NOT NULL necessari en el procediment que no tingui valor [en aquest cas el identificador] o que existeix el identificador) i si tot es correcte es modifica el registre (dFiCatCast) que indica la data en que finalitza la categoria del casteller i s'envia un missatge, a través de la variable de sortida, de procés satisfactori. En cas contrari, es genera un error que també es transmet pel paràmetre de sortida

Errors:

1. E_campNull: Un camp NULL no té valor
2. E_ccNoExistent: No existeix el registre que es passa per paràmetre
3. E_data: La data fi ha de ser posterior a la data inicial

4.10. Triggers

S'han programat diferents triggers encarregats d'incrementar el valor dels identificadors dels registres de les taules cada vegada que es faci una nova inserció. Així doncs, de forma totalment automàtica, obtindrem el següent:

```

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_CASTELL
BEFORE INSERT ON castell FOR EACH ROW
BEGIN
  SELECT SQ_idCastell.NEXTVAL
    INTO :NEW.idCastell
    FROM DUAL;
END TR_BEF_INS_CASTELL;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_ACTUACIO
BEFORE INSERT ON actuacio FOR EACH ROW
BEGIN
  SELECT SQ_idActuacio.NEXTVAL
    INTO :NEW.idActuacio
    FROM DUAL;
END TR_BEF_INS_ACTUACIO;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_CASTELLSACTUACIO
BEFORE INSERT ON castellsActuacio FOR EACH ROW
BEGIN
  SELECT SQ_idCastAct.NEXTVAL
    INTO :NEW.idCastAct
    FROM DUAL;
END TR_BEF_INS_CASTELLSACTUACIO;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_EXTERN
BEFORE INSERT ON extern FOR EACH ROW
BEGIN
  SELECT SQ_idExtern.NEXTVAL
    INTO :NEW.idExtern
    FROM DUAL;
END TR_BEF_INS_EXTERN;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_PERSONA
BEFORE INSERT ON persona FOR EACH ROW
BEGIN
  SELECT SQ_idPersona.NEXTVAL
    INTO :NEW.idPersona
    FROM DUAL;
END TR_BEF_INS_PERSONA;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_FACTURA
BEFORE INSERT ON factura FOR EACH ROW
BEGIN

```

```
SELECT SQ_idFactura.NEXTVAL
  INTO :NEW.idFactura
  FROM DUAL;
END TR_BEF_INS_FACTURA;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_CATEGORIA
BEFORE INSERT ON categoria FOR EACH ROW
BEGIN
  SELECT SQ_idCategoria.NEXTVAL
    INTO :NEW.idCategoria
    FROM DUAL;
END TR_BEF_INS_CATEGORIA;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_CASTELLER
BEFORE INSERT ON casteller FOR EACH ROW
BEGIN
  SELECT SQ_idCasteller.NEXTVAL
    INTO :NEW.idCasteller
    FROM DUAL;
END TR_BEF_INS_CASTELLER;
/

CREATE OR REPLACE TRIGGER XIQUETS.TR_BEF_INS_CATEGORIACASTELLER
BEFORE INSERT ON categoriaCasteller FOR EACH ROW
BEGIN
  SELECT SQ_idCatCast.NEXTVAL
    INTO :NEW.idCatCast
    FROM DUAL;
END TR_BEF_INS_CATEGORIACASTELLER;
/
```

5. Joc de proves

Per tal de comprovar el correcte funcionament de la base de dades, s'han implementat dos scripts que utilitzaran les funcionalitats anteriorment proporcionades

Abans d'executar el fitxer del joc de proves, serà necessari omplir les taules de la base de dades. Per això s'executarà el fitxer "10. CargaDades" i, d'aquesta forma, tindrem diferents registres a les nostres taules.

Una vegada s'hagin carregat les dades passarem a realitzar el joc de proves. Aquest consistirà en introduir, modificar i eliminar registres. D'aquesta forma observarem que es compleixen les restriccions especificades com a requeriments de les nostres funcionalitats.

En aquest test doncs, que podrem trobar dins del fitxer anomenat "11.JocProves", es realitzaran proves "buscant" l'error del sistema per tal de comprovar el bon funcionament de la base de dades.

Les proves que es realitzaran son les següents:

- Castell
 1. Realitzem una alta d'un castell correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'un castell que ja esta donat d'alta: Intentem donar d'alta un registre amb un nom que ja existeix. Aquesta situació ens donarà un error
 3. Introduïm que els punts d'un castell carregat son superiors que els punts d'un castell descarregat: Introduïm un numero superior a la dada punts carregat envers la dada punts descarregat (fet no real). El sistema genera un error
 4. Modifiquem un castell correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 5. Modifiquem un castell que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error

6. Modifiquem un castell i el resultat es dos castells iguals: Modifiquem la dada nom i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error
 7. Modifiquem els punts, quedant que els punts d'un castell carregat son superiors que els punt d'un castell descarregat: Modifiquem les dades de punts (carregat i/o descarregat) i no es compleix la condició punts descarregat > punts carregat. El sistema genera un error
 8. Donem de baixa un castell correctament: Realitzem la baixa lògica d'un registre. No l'eliminem de la base de dades, si no que modifiquem la dada baixacastell
 9. Donem de baixa un castell que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error
 10. Donem de baixa un castell que ja estava donat de baixa: Introduïm un identificador que ja havia estat donat de baixa (baixacastell = 'S'). El sistema genera un error
- Actuació
 1. Realitzem una alta d'una actuació correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'una actuació que ja esta donada d'alta: Intentem donar d'alta un registre amb un nom, una data i una població que ja existeix. Aquesta situació ens donarà un error
 3. Modifiquem una actuació correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 4. Modifiquem una actuació que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error
 5. Modifiquem una actuació i el resultat es dos actuacions iguals: Modifiquem les dades nom, data i població (una d'elles o totes) i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error
 6. Donem de baixa una actuació correctament: Realitzem la baixa lògica d'un registre. No l'eliminem de la base de dades, si no que modifiquem la dada baixaactuacio
 7. Donem de baixa una actuació que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error

8. Donem de baixa una actuació que ja estava donada de baixa: Introduïm un identificador que ja havia estat donat de baixa (baixaactuacio = 'S'). El sistema genera un error
- CastellsActuacio
 1. Realitzem una alta d'un castell a una actuació correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'un castell a una actuació que ja esta donat d'alta: Intentem donar d'alta un registre amb un identificador (d'actuació i de castell) amb un estat (de castell) que ja existeix. Aquesta situació ens donarà un error
 3. Realitzem una alta d'un castell a una actuació, d'un castell que no existeix: Introduïm un identificador (de castell) que no existeix dins de la base de dades. El sistema genera un error
 4. Realitzem una alta d'un castell a un actuació, en una actuació que no existeix: Introduïm un identificador (d'actuació) que no existeix dins de la base de dades. El sistema genera un error
 5. Modifiquem un castell d'una actuació correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 6. Modifiquem un castell d'una actuació que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error
 7. Modifiquem un castell d'una actuació i el resultat es dos castells a una actuació iguals: Modifiquem la dada estat d'un registre i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error.
 8. Donem de baixa un castell d'una actuació correctament: Realitzem la baixa física d'un registre. L'eliminem de la base de dades
 9. Donem de baixa un castell d'una actuació que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error
 - Categoria
 1. Realitzem una alta d'una categoria correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'una categoria que ja esta donada d'alta: Intentem donar d'alta un registre amb un nom que ja existeix. Aquesta situació ens donarà un error

3. Modifiquem una categoria correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 4. Modifiquem una categoria que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error
 5. Modifiquem una categoria i el resultat es dos categories iguals: Modifiquem la dada nom i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error
 6. Donem de baixa una categoria correctament: Realitzem la baixa lògica d'un registre. No l'eliminem de la base de dades, si no que modifiquem la dada baixacategoria
 7. Donem de baixa una categoria que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error
 8. Donem de baixa una categoria que ja estava donada de baixa: Introduïm un identificador que ja havia estat donat de baixa (baixacategoria = 'S'). El sistema genera un error
- Casteller
 1. Realitzem una alta d'un casteller correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'un casteller que ja esta donat d'alta: Intentem donar d'alta un registre amb un DNI que ja existeix. Aquesta situació ens donarà un error
 3. Modifiquem un casteller correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 4. Modifiquem un casteller que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error
 5. Modifiquem un casteller i el resultat es dos castellers iguals: Modifiquem la dada DNI i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error
 6. Donem de baixa un casteller correctament: Realitzem la baixa lògica d'un registre. No l'eliminem de la base de dades, si no que modifiquem la dada baixacasteller. També podem modificar la dada motiubaixa
 7. Donem de baixa un casteller que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error

8. Donem de baixa un casteller que ja estava donat de baixa: Introduïm un identificador que ja havia estat donat de baixa (baixacasteller = 'S'). El sistema genera un error
- CategoriaCasteller
 1. Realitzem una alta d'una categoria d'un casteller correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'una categoria d'un casteller que ja esta donada d'alta: Intentem donar d'alta un registre amb un identificador (de categoria i de casteller) amb una data que ja existeix. Aquesta situació ens donarà un error
 3. Realitzem una alta d'una categoria d'un casteller, d'una categoria que no existeix: Introduïm un identificador (de categoria) que no existeix dins de la base de dades. El sistema genera un error
 4. Realitzem una alta d'una categoria d'un casteller, a un casteller que no existeix: Introduïm un identificador (de casteller) que no existeix dins de la base de dades. El sistema genera un error
 5. Modifiquem una categoria d'un casteller correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 6. Modifiquem una categoria d'un casteller que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error
 7. Modifiquem una categoria d'un casteller i el resultat es dos categories a un casteller iguals: Modifiquem la dada nom i/o data inici i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error.
 8. Donem de baixa una categoria d'un casteller correctament: Modifiquem la dada datafi amb la data en que finalitza la categoria. En aquest cas no es realitza cap baixa estrictament parlant (ni lògica ni física)
 9. Donem de baixa una categoria d'un casteller actuació que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error
 10. Donem de baixa una categoria d'un casteller amb la data final mes "aviat" que la data inicial
 - Extern

1. Realitzem una alta d'un extern correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'un extern que ja esta donat d'alta: Intentem donar d'alta un registre amb un CIF que ja existeix. Aquesta situació ens donarà un error
 3. Modifiquem un extern correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 4. Modifiquem un extern que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error
 5. Modifiquem un extern i el resultat es dos externs iguals: Modifiquem la dada CIF i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error
 6. Donem de baixa un extern correctament: Realitzem la baixa lògica d'un registre. No l'eliminem de la base de dades, si no que modifiquem la dada baixaextern
 7. Donem de baixa un extern que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error
 8. Donem de baixa un extern que ja estava donat de baixa: Introduïm un identificador que ja havia estat donat de baixa (baixaextern = 'S'). El sistema genera un error
- Persona
 1. Realitzem una alta d'una persona correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'una persona que ja esta donada d'alta: Intentem donar d'alta un registre amb un nom i cognoms que ja existeix. Aquesta situació ens donarà un error
 3. Realitzem una alta d'una persona a un extern que no existeix: Introduïm un identificador (d'extern) que no existeix dins de la base de dades. El sistema genera un error
 4. Modifiquem una persona correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 5. Modifiquem una persona que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error.
 6. Modifiquem una persona i el resultat es dos persones iguals: Modifiquem la dada nom i la dada cognoms i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error.

7. Donem de baixa una persona correctament: Realitzem la baixa física d'un registre. L'eliminem de la base de dades.
 8. Donem de baixa una persona que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error
- Factura
 1. Realitzem una alta d'una factura correctament: Introduïm totes les dades NOT NULL necessàries per tal de que l'alta es faci correctament
 2. Realitzem una alta d'una factura que ja esta donada d'alta: Intentem donar d'alta un registre amb una referencia que ja existeix. Aquesta situació ens donarà un error
 3. Realitzem una alta d'una factura a un extern que no existeix: Introduïm un identificador (d'extern) que no existeix dins de la base de dades. El sistema genera un error
 4. Modifiquem una factura correctament: Introduïm les dades que volem modificar sense que aquestes provoquin un conflicte
 5. Modifiquem una factura que no existeix: Introduïm un identificador que no existeix dins de la base de dades. El sistema genera un error.
 6. Modifiquem una factura i el resultat es dos factures iguals: Modifiquem la dada referencia i el contingut és el mateix que un altre registre, situació que ens suposaria tenir dos registres iguals. El sistema genera un error.
 7. Donem de baixa una factura correctament: Realitzem la baixa física d'un registre. L'eliminem de la base de dades.
 8. Donem de baixa una factura que no existeix: Introduïm un identificador que no existeix a la base de dades. El sistema genera un error

El resultat de totes les proves esmentades anteriorment, vindrà donat per la variable RSP (sortida) de cada procediment. Aquesta variarà segons la prova que haguem realitzat i d'aquesta forma comprovarem si el resultat s'adapta, com hem dit anteriorment, als requisits funcionals.

6. Conclusions

Vam començar aquest projecte motivat per dues situacions: la primera, la necessitat de realitzar un treball per a finalitzar els estudis d'enginyeria, i la segona, la possibilitat de donar un cop de mà a una entitat sense cap ànim de lucre i, com les demés colles castelleres, no professionalitzada. Una mescla perfecte i motivadora.

Durant el procés de realització del treball, hem pres consciència de que no és el mateix dur a terme un projecte en base a un enunciat, a haver d'incloure l'enunciat dins del mateix procés de creació. Sabíem quina era la finalitat (realitzar una BBDD) però no sabíem quines eren les necessitats que aquesta havia de resoldre. Així doncs, l'enunciat ha estat extret de les diverses reunions entre el client i la part desenvolupadora, situació que, al nostre entendre, ha complicat la realització del projecte però que, per un altre costat la fet més real.

En conseqüència, hem pogut comprovar la dificultat i l'essencialitat de realitzar una bona programació (marcada per les dates exigides per la universitat) de les diverses fases que componen un projecte. En aquest punt, tal i com comentàvem al paràgraf anterior, hem experimentat que la situació de reunir-se amb el client (inexpert i sense cap coneixement tècnic, com marquen els diferents llibres de gestions informàtiques) és de les més importants i complicats, ja que et marca que la feina estigui mal o ben feta.

Per un altre costat, dins del procés de disseny i implementació del SGBD, hem pogut veure que l'adaptació a nivell informàtic de les necessitats expressades pel client, no es poden plasmar directament com si féssim un dibuix, si no que, amb l'ajut de les diferents regles de disseny i creació de les bases de dades, s'ha de buscar la 'fórmula' per encabir-hi totes les peticions i que l'estructura tingui integritat i coherència.

Per finalitzar aquest apartat, volem expressar que vam començar el projecte pensant que el producte seria un 'tot' i que, com a molt, el projecte tindria algunes modificacions i/o actualitzacions, però durant el desenvolupament del mateix, hem comprovat que una base de

dades, avui en dia, té i esperem que en aquest cas tingui, moltes més utilitats que la simple funció d'emmagatzematge d'informació. Per tant, entenem i esperem que aquesta base de dades serveixi com a base per al desenvolupament de futures aplicacions o altres programes que siguin beneficioses per al client.

7. Bibliografia

7.1. Llibres de consulta

- “Introducción a los Sistemas de Bases de Datos” C.J. DATE
- Apunts de les assignatures de la UOC
 - Informàtica aplicada a la gestió
 - Bases de dades 1
 - Bases de dades 2
 - Enginyeria del programari
 - Tècniques de desenvolupament de programari

7.2. Fonts d'internet

- Funcionament d'oracle <http://psoug.org/reference/packages.html>
 - Paquets
 - Seqüències
 - Usuari
 - Tablespace
 - Triggers