



# Sistema de gestió de base de dades per les competicions de la Federació Internacional d'Automobilisme

**Ferran Moreno Núñez**  
Grau Enginyeria Informàtica

**Consultor: Jordi Ferrar Duran**

Data Lliurament: 15 de Juny de 2015



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

**Llicències alternatives (triari alguna de les seqüents i substituir la de la pàgina anterior)**

**A) Creative Commons:**



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-SenseObraDerivada 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement-CompartirIgual 3.0 Espanya de Creative Commons](#)



Aquesta obra està subjecta a una llicència de [Reconeixement 3.0 Espanya de Creative Commons](#)

**B) GNU Free Documentation License (GNU FDL)**

Copyright © ANY EL-TEU-NOM.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant

Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

### **C) Copyright**

© (l'autor/a)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

## FITXA DEL TREBALL FINAL

<b>Títol del treball:</b>	<i>Sistema de gestió de base de dades per les competicions de la Federació Internacional d'Automobilisme</i>
<b>Nom de l'autor:</b>	<i>Ferran Moreno Núñez</i>
<b>Nom del consultor:</b>	<i>Jordi Ferrar Duran</i>
<b>Data de lliurament:</b>	<i>06/2015</i>
<b>Àrea del Treball Final:</b>	<i>Bases de dades</i>
<b>Titulació:</b>	<i>Grau Enginyeria Informàtica</i>

### **Resum del Treball (màxim 250 paraules):**

Aquest treball tracte de donar resposta a la necessitat de crear una eina per a enregistrar la activitat que du a terme la FIA. Això implica poder desar les dades referents a les entitats que formen part de la associació (persones, equips, patrocinadors, emissores, etc...) així com aquelles entitats necessàries per a representar les competicions (com per exemple, circuits, categories, o dates). Una part important de la solució és el mòdul de telemetria, en el qual es volen desar les dades telemètriques generades pels cotxes de competició durant les curses.

Per al tractament de dades s'han creat una sèrie de procediments d'ABM que permeten inserir, esborrar i modificar dades de manera encapsulada. Aquests procediments són també responsables de la validació de les dades que s'insereixen, compten amb un tractament d'errors i actualitzen una taula de *log* que monitoritza les accions executades a la base de dades.

La segona part del treball és un repositori estadístic referent a les cerques que el client realitza més sovint. Aquest es manté actualitzat en tot moment ja que són els propis procediments d'ABM els qui actualitzen el repositori després de cada modificació a la base de dades. Com a funcionalitat addicional, s'han implementat una sèrie de procediments que permeten actualitzar el repositori independentment dels procediments d'ABM i en qualsevol moment, basant-se en les dades enregistrades al sistema.

Per avaluar el sistema i com a funcionalitat addicional, s'han creat certs procediments per a generar dades telemètriques realistes aleatòriament.

**Abstract (in English, 250 words or less):**

With this project it is intended to solve the necessity of the FIA of having an environment which allows them to store and register their activity. This means being able to store data referring to entities involved in the association (people, teams, sponsors, broadcasters, etc...) and also the necessary objects to represent the competitions (i.e. circuits, dates or categories). One of the top features of the solution is the telemetry support, which is meant to store and analyze the data generated by the devices on the car during the races.

CRUD procedures are responsible for data management, allowing the user to insert, delete and modify entries on an encapsulated manner. They are also responsible for validating data, error handling and updating the log table that monitors each procedure execution over the database.

The second module of the project is about a data warehouse that generates statistics regarding the most common queries used by the customer. It is always up to date since the same CRUD procedures are responsible for updating it after every execution. As an additional feature, the solution provides several procedures that allow the data warehouse to be updated beside the CRUD procedures at any moment based on the available data within the database tables.

As an addition and for testing purposes, the solution comes with some procedures that generate the proper amount of random data for the telemetry module.

**Paraules clau (entre 4 i 8):**

Database, Data warehouse, MySQL, Relational model, Procedures

# Índex

1. <i>Introducció</i> .....	1
<b>1.1 Context i justificació del Treball</b> .....	1
<b>1.2 Objectius del Treball</b> .....	1
<b>1.3 Enfocament i mètode seguit</b> .....	1
<b>1.4 Planificació del Treball</b> .....	2
<b>1.5 Breu sumari de productes obtinguts</b> .....	2
<b>1.6 Breu descripció dels altres capítols de la memòria</b> .....	3
2. <i>Planificació</i> .....	4
<b>2.1 Metodologia de treball</b> .....	4
<b>2.2 Pla de treball</b> .....	5
<b>2.2.1 PAC 1</b> .....	6
<b>2.2.2 PAC 2</b> .....	9
<b>2.2.3 PAC 3</b> .....	11
<b>2.2.4 Lliurament final</b> .....	13
<b>2.3 Gestió del temps</b> .....	14
<b>2.3.1 PAC 1</b> .....	14
<b>2.3.2 PAC 2</b> .....	14
<b>2.3.3 PAC 3</b> .....	15
<b>2.3.4 Lliurament final</b> .....	15
3. <i>Disseny conceptual</i> .....	17
<b>3.1 Anàlisi de requisits</b> .....	17
<b>3.2 Definició detallada d'entitats i atributs</b> .....	18
<b>3.3 Relacions entre les entitats</b> .....	23
<b>3.4 Diagrama UML</b> .....	27
4. <i>Disseny lògic</i> .....	28
<b>4.1 Transformació del model conceptual al model relacional</b> .....	28
<b>4.2 Normalització del disseny [17] [18]</b> .....	31
5. <i>Disseny físic</i> .....	33
<b>5.1 Transformació del model lògic al model físic</b> .....	33
<b>5.2 Procediments d'ABM i disparadors</b> .....	33

5.3	<b>Log i control d'errors.</b>	53
6.	<i>Mòdul estadístic</i>	56
6.1	<b>Disseny conceptual</b>	56
6.2	<b>Disseny físic</b>	60
7.	<i>Test</i>	67
8.	<i>Conclusions</i>	69
9.	<i>Glossari</i>	70
10.	<i>Bibliografia</i>	72



## Llista de figures

Fig. 2.1 Procés de la metodologia en cascada.....	5
Fig. 2.2 Diagrama Gantt de la PAC 1 .....	7
Fig. 2.3 Diagrama Gantt del lliurament final .....	7
Fig. 2.4 Diagrama Gantt de la PAC 2 .....	8
Fig. 2.5 Diagrama Gantt de la PAC 3 .....	8
Fig. 3.1 Entitat Competició en format UML.....	18
Fig. 3.2 Entitat Sector en format UML .....	18
Fig. 3.3 Entitat País en format UML .....	19
Fig. 3.4 Entitat Persona en format UML .....	19
Fig. 3.5 Entitat Equip en format UML .....	19
Fig. 3.6 Entitat Director en format UML.....	20
Fig. 3.7 Entitat Cotxe en format UML .....	20
Fig. 3.8 Entitat Patrocinador en format UML .....	20
Fig. 3.9 Entitat Fabricant en format UML.....	21
Fig. 3.10 Entitat Component en format UML .....	21
Fig. 3.11 Entitat Circuit en format UML .....	22
Fig. 3.12 Entitat Emissora en format UML.....	22
Fig. 3.13 Entitat Pilot en format UML.....	22
Fig. 3.14 Entitat Telemetria en format UML.....	23
Fig. 3.15 Relació Patrocini entre les entitats Equip i Patrocinador. ....	23
Fig. 3.16 Relació Contractat entre Equip i Pilot. ....	24
Fig. 3.17 Relació Patrocini entre Equip i Director. ....	24
Fig. 3.18 Relació Té entre les entitats Equip i Cotxe.....	24
Fig. 3.19 Relació entre Circuit, Pilot i Competició al voltant de Cursa.....	25
Fig. 3.20 Relacions Utilitza entre Cotxe i Component, i Proporciona entre Fabricant i Component. ....	25
Fig. 3.21 Relació Genera entre Telemetria i Component. ....	26
Fig. 3.22 Relació de generalització entre Persona, Director i Pilot.....	26
Fig. 3.23: Relació Retransmissió entre les entitats Emissora i Competició. ....	26
Fig. 3.24: Relació entre les entitats Persona, Circuit i Emissora vers País .....	27
Fig. 3.25 Diagrama UML complet.....	27
Fig. 5.1 Paràmetres del procediment alta_circuit .....	34

Fig. 5.2 Paràmetres del procediment alta_competició.....	34
Fig. 5.3 Paràmetres del procediment alta_component.....	35
Fig. 5.4 Paràmetres del procediment alta_cotxe .....	35
Fig. 5.5 Paràmetres del procediment alta_cursa .....	36
Fig. 5.6 Paràmetres del procediment alta_director.....	36
Fig. 5.7 Paràmetres del procediment alta_emissora .....	36
Fig. 5.8 Paràmetres del procediment alta_equip.....	37
Fig. 5.9 Paràmetres del procediment alta_fabricant .....	37
Fig. 5.10 Paràmetres del procediment alta_país .....	37
Fig. 5.11 Paràmetres del procediment alta_patrocinador.....	38
Fig. 5.12 Paràmetres del procediment alta_patrocini .....	38
Fig. 5.13 Paràmetres del procediment alta_pilot .....	39
Fig. 5.14 Paràmetres del procediment alta_retransmissió .....	39
Fig. 5.15 Paràmetres del procediment alta_sector .....	39
Fig. 5.16 Paràmetres del procediment alta_telemetria .....	40
Fig. 5.17 Paràmetres del procediment alta_temporada.....	40
Fig. 5.18 Paràmetres del procediment baixa_circuit.....	40
Fig. 5.19 Paràmetres del procediment baixa_competició .....	41
Fig. 5.20 Paràmetres del procediment baixa_component .....	41
Fig. 5.21 Paràmetres del procediment baixa_cotxe .....	41
Fig. 5.22 Paràmetres del procediment baixa_cursa .....	42
Fig. 5.23 Paràmetres del procediment baixa_director .....	42
Fig. 5.24 Paràmetres del procediment baixa_emissora .....	42
Fig. 5.25 Paràmetres del procediment baixa_equip .....	42
Fig. 5.26 Paràmetres del procediment baixa_fabricant .....	43
Fig. 5.27 Paràmetres del procediment baixa_país .....	43
Fig. 5.28 Paràmetres del procediment baixa_patrocinador.....	43
Fig. 5.29 Paràmetres del procediment baixa_patrocini .....	44
Fig. 5.30 Paràmetres del procediment baixa_pilot .....	44
Fig. 5.31 Paràmetres del procediment baixa_retransmissió.....	44
Fig. 5.32 Paràmetres del procediment baixa_sector .....	45
Fig. 5.33 Paràmetres del procediment baixa_telemetria .....	45
Fig. 5.34 Paràmetres del procediment baixa_temporada.....	45
Fig. 5.35 Paràmetres del procediment modificació_circuit .....	46

Fig. 5.36 Paràmetres del procediment modificació_competició.....	46
Fig. 5.37 Paràmetres del procediment modificació_component.....	46
Fig. 5.38 Paràmetres del procediment modificació_cotxe .....	47
Fig. 5.39 Paràmetres del procediment modificació_cursa.....	47
Fig. 5.40 Paràmetres del procediment modificació_director.....	48
Fig. 5.41 Paràmetres del procediment modificació_emissora .....	48
Fig. 5.42 Paràmetres del procediment modificació_equip.....	49
Fig. 5.43 Paràmetres del procediment modificació_fabricant.....	50
Fig. 5.44 Paràmetres del procediment modificació_país .....	50
Fig. 5.45 Paràmetres del procediment modificació_patrocinador.....	51
Fig. 5.46 Paràmetres del procediment modificació_patrocini .....	51
Fig. 5.47 Paràmetres del procediment modificació_pilot .....	51
Fig. 5.48 Paràmetres del procediment modificació_sector.....	52
Fig. 5.49 Paràmetres del procediment modificació_telemetria .....	52
Fig. 5.50 Paràmetres del procediment modificació_temporada.....	53
Fig. 5.51 Representació UML de la taula LOG.....	54
Fig. 6.1 Representació UML de la taula de fets.....	58

# 1. Introducció

## 1.1 Context i justificació del Treball

Finalment, i després d'anys d'estudi, arriba el moment de plasmar els coneixements adquirits per mitjà del treball de fi de grau. L'àmbit de les bases de dades sempre m'ha resultat atractiu, la seva necessitat a l'àmbit de les TIC és innegable i és sempre un gran repte crear un bon disseny i veure com es fa realitat.

En aquest cas el projecte tracte de donar resposta a la necessitat de la FIA d'enregistrar totes les dades necessàries en referència a les entitats relacionades amb l'associació. Això és indispensable ja que en una entitat d'aquestes característiques cal controlar moltes dades per a garantir el seu bon funcionament. Per exemple, s'ha de portar un control de les competicions, dels participants, dels resultats i dels vehicles entre d'altres. El plantejament d'aquest projecte és un bon exemple de la utilitat de les bases de dades a l'actualitat, ja que fa paler que qualsevol activitat amb certa complexitat i duració en el temps necessita un sistema d'aquest tipus.

És especialment digne de menció que el treball pretén, a part de crear un disseny adequat al problema per a desar les dades, crear un repositori estadístic que permeti obtenir informació complexa amb temps d'accés mínim. La creació d'un magatzem de dades amb aquesta finalitat que funcioni de manera eficient serà un dels objectius claus d'aquest projecte, així com aconseguir que totes les funcions implementades gaudeixin de robustesa i control d'errors.

## 1.2 Objectius del Treball

El meu objectiu principal és adquirir coneixements que em puguin ajudar a l'àmbit professional així com perfeccionar aquells coneixements dels que actualment gaudeixo. Després d'haver desenvolupat diverses funcions a la meua carrera trobo especialment interessant l'àmbit de les bases de dades, i no és cap utopia que MySQL és una tecnologia molt estesa i en procés d'expansió. Per això penso que tot el temps dedicat a aquest treball serà de gran utilitat per a futur projectes.

## 1.3 Enfocament i mètode seguit

Enceto aquesta tasca des de la modèstia professional. No tinc cap referència personal per a comparar o fer servir com a model de base per a aquest treball, ja que tots aquells projectes similars en què he treballats o ja estaven creats o eren de caire més senzill.

He començat seguint el model descrit a l'assignatura "disseny de bases de dades", el materials de la qual han resultat molt útils al llarg del desenvolupament del treball. Com és habitual en

projectes realitzats amb manca ferma d'experiència, he comés diversos errors de disseny que m'han obligat a redissenya la solució més de dues vegades (quan la meva intenció era realitzar un plantejament de desenvolupament en cascada). Finalment he encetat el projecte des de l'inici intentant a cada fase cometre els mínims errors per a no haver de tornar enrere. És l'estratègia no més eficient però si la idònia per a la meva situació, ja que qualsevol altre manera de treballar es veuria afectada per la meva manca d'experiència. D'aquesta manera he pogut aprendre dels meus errors, tot i que el projecte en si mateix s'ha vist afectat per la manca de temps produïda pels errors comesos.

## 1.4 Planificació del Treball

La planificació del projecte s'ha estructurant basant-se en les dates d'entrega de les pacs així com tenint en compte els continguts bàsics que cadascuna d'aquestes entregues han de contenir. S'ha escollit la metodologia en cascada per a la seva implementació i s'han considerat els possibles riscos que poden sorgir en aquest temps.

Donada la importància i mida d'aquest tema s'ha considerat oportú crear un capítol específic per a la seva descripció (veure capítol Planificació).

## 1.5 Breu sumari de productes obtinguts

El producte final obtingut representa un sistema de gestió per a la FIA amb tots els requisits establerts a l'enunciat. Sóc conscient de que, tot i que sembla un tòpic a la gestió de projectes, el temps del què he disposat ha estat insuficient per a realitzar un producte totalment fiable i optimitzat. Dit això, el projecte compte amb tots els mòduls requerits alguns de complementaris. Aquests són:

- Model relacional de taules i atributs necessaris per a donar resposta als requeriments del client
- Procediments d'alta, baixa i modificació (ABM) per a administrar les dades del sistema.
- Control d'errors al procediments que pren control dels sistema d'errors per defecte per tal de poder proporcionar missatges personalitzats al paràmetre de sortida .
- Sistema de *log* que registre les accions realitzades pels procediments ABM.
- Estructura del repositori estadístic consistent en una taula de fets i les taules complementàries,
- Disparadors per controlar que els arguments d'entrada són coherents.
- Procediments d'ABM adaptats per a actualitzar les estadístiques incrementalment.
- **Procediments addicionals** que generen la majoria de les estadístiques partint d'un estat inicial amb dades al sistema que no han estat afegides de manera incremental al mòdul estadístic.
- **Procediments addicionals** que insereixen dades aleatòries (però raonables) a la taula telemetria per tal de comptar amb una quantitat adient de dades que representi una situació real.

## 1.6 Breu descripció dels altres capítols de la memòria

El treball esta estructurat en sis capítols que fan referencia al procés d'implementació. A continuació es descriu cada capítol així com la seva relació amb el producte global.

Inicialment es defineix, al capítol "Planificació", quina es la metodologia de treball escollida, el pla de treball que es seguirà així com un anàlisi del temps emprat per a cada tasca i els riscos associats.

Els tres capítols següents descriuen el procés d'implementació de la base de dades relacional. Inicialment es du a terme el disseny conceptual, que inclou l'anàlisi de requisits així com la creació d'un diagrama UML que proporcioni les taules amb els atributs necessaris per tal de donar suport a la posterior implementació de la solució. A continuació, es transforma aquest disseny conceptual en un disseny lògic, definint les claus foranes i primàries per a cada taula així com el tipus de dades que es faran servir dependents amb la entitat que representen. El disseny lògic finalitza amb la normalització del disseny obtingut per tal d'optimitzar-lo i trobar possibles errors o inconsistències. Al tercer capítol es descriu la transformació del model lògic a model físic, que consisteix en la creació de les taules, la implementació dels procediments d'alta, baixa i modificació així com dels disparadors, la creació d'un sistema de *logs* i la implementació d'un sistema de control d'errors.

Un cop finalitzada la descripció de la base de dades relacional, el sisè capítol descriu la creació del magatzem de dades. Aquest capítol inclourà descripció de totes les fases del disseny d'aquest mòdul: s'analitza quines son les estadístiques que s'han de considerar, es dissenya una estructura de taula de fets i taules complementaries, s'actualitzen els procediments existents per a mantenir les estadístiques actualitzades en tot moment, i s'implementen un sèrie de procediments addicionals per permetre generar estadístiques amb les dades existents al sistema en qualsevol moments.

El treball d'implementació finalitza amb el capítol dedicat a les proves efectuades sobre la solució final. Es proporciona una breu descripció de les tasques efectuades per a validar el disseny així com quins son els arxius annexes fets servir per tal de provar la funcionalitat del sistema.

Al final de la memòria es poden trobar les conclusions obtingudes un cop finalitzat el projecte, un glossari dels termes mes habituals utilitzats en aquest document així com la bibliografia que s'ha fet servir al llarg del projecte.

## 2. Planificació

El primer pas i potser dels més importants d'un projecte de llarga durada com aquest és la definició del pla de treball, la definició del qual és l'objectiu d'aquest apartat. La meua situació actual a canviat dràsticament als darrers mesos, ja que he encetat un nou projecte professional i personal a un altre país. Això comporta que el temps del que dispo per a realitzar aquest projecte és limitat, i fa encara més necessària una bona gestió dels recursos disponibles.

Seguidament s'intentaran definir de manera acurada els següents punts:

- Metodologia que es farà servir.
- Definició de tasques, material necessari i documentació a lliurar.
- Gestió del temps i de possibles incidències.

### 2.1 Metodologia de treball

Els projectes de disseny i implementació de bases de dades relacionals tenen certes regles i indicadors que ens poden ajudar a escollir la metodologia més adient a seguir. Normalment un projecte d'aquest tipus consta de les fases següents:

- Anàlisi del problema i la recollida de requisits
- Definició del disseny conceptual.
- Definició del disseny lògic
- Definició del disseny físic.
- Implementació al sistema de gestió de bases de dades.

Escollir una metodologia en cascada implica que cadascuna de les fases requereix que la fase anterior hagi finalitzat per a poder començar. És un sistema més rígid en el qual no es contemplen canvis en la definició del problema un cop encetat el projecte, i per tant, si al final es produïssin, el temps necessari per a realitzar les modificacions pot ser elevat. Si el projecte per definició és incomplet o es preveu diversos canvis al disseny al llarg del projecte, la metodologia àgil ofereix una manera més practica de dur a terme la tasca. Aquest sistema preveu diverses iteracions completes al llarg del projecte i és per això que els canvis al disseny no resulten tan crítics. En la nostre situació l'enunciat està ben definit, l'objectiu és clar i les fites temporals són clares. Una metodologia en cascada és, per tant, la opció més recomanada, ja que ens permetrà definir més clarament els objectius i terminis sense haver dedicat recursos a preveure possibles canvis al disseny posteriori.

La metodologia en cascada consta de les següents fases, les quals segueixen una lògica similar a les etapes d'un projecte de bases de dades descrites prèviament:

- Anàlisi de requisits.
- Disseny del sistema.
- Disseny del programa.
- Implementació.
- Proves.
- Verificació.

Aquesta metodologia es basa en la documentació per tal d'assegurar que cadascuna de les fases ha estat finalitzada amb èxit. Per això, al finalitzar cadascuna de les etapes, caldrà redactar un informe per definir el que s'ha fet a la fase prèvia i que tots els requisits necessaris per a encetar la següent fase estan preparats.



Fig. 2.1 Procés de la metodologia en cascada.

## 2.2 Pla de treball

El projecte està dividit pels lliuraments de les proves d'avaluació contínua en quatre blocs (pla de treball, pac 2, pac 3 i lliurament final) es planificarà el projecte fent servir les dates d'entrega com a guia. D'aquesta manera, la definició inicial de tasques a realitzar al llarg del curs així com les fites relacionades es descriuen als punts següents.

El temps del que dispenso per a realitzar el projecte és limitat. Pràcticament, de dilluns a dimecres no tindrè cap estona disponible. Aquest fet és representat a la planificació, de tal manera que intentaré compensar aquesta mancança amb cap de setmanes y tardes lliures.

Hauria volgut considerar molt més la gestió d'imprevistos, però durant la confecció del calendari he comprès que les hores queden molt ajustades. Quan era raonable he comptat 1 o 2 dies per imprevistos d'última hora a cada PAC, però si es perdessin encara més dies dedicats al TFG, començaria a dubtar que un treball adient es pogués dur a terme.

En quant a la documentació, caldrà anar confeccionant un esborrany de memòria després de cada apartat, així com redactar un informe un cop finalitzat un capítol per asseverar que els prerequisits del següent punt estan coberts abans de començar. Aquesta documentació es lliurarà de manera conjunta amb cada pac si aporta contingut, o es farà referència al seu contingut si es tracta d'alguna part de la memòria en procés.



## 2.2.1 PAC 1

Aquest primer bloc té per objectiu l'entrega de la documentació relacionada amb el pla de treball que es seguirà al llarg del projecte. Tot i que les fites han estat medidades, es planteja la possibilitat de modificar aquest pla després de cada lliurament, ja sigui per afegir noves tasques que sorgeixin al transcurs del semestre com per variar la temporització segons el cas.

De cara a crear una documentació inicial el més acurada possible i de preparar les eines necessàries per a encetar el segon bloc de la millor manera, les tasques previstes per aquestes setmanes són les següents:

### 2.2.1.1 Anàlisi de requisits

[ Duració: 1 dies ]

Inicialment s'analitzarà la situació per a poder planificar la realització del projecte de manera coherent. Per això serà necessari:

- Llegir els documents facilitats a l'aula de l'assignatura, que són l'enunciat del TFG i les recomanacions per a la realització del TFG.
- També caldrà rellegir els apunts de les assignatures de Disseny de bases de dades i d'Enginyeria del programari per tal de tenir presents les diferents metodologies de treball existents així com l'estructura d'un projecte d'aquest tipus.
- Serà necessari cercar informació sobre la creació a Oracle dels *data warehouse* i del sistema de *logs*, per tal de definir quin esforç serà necessari per al seu desenvolupament.
- Per últim, s'analitzarà quin programari serà necessari instal·lar per a realitzar el projecte i la documentació associada.

### 2.2.1.2 Instal·lació del programari

[ Duració: 1 dia - Predecessora: Anàlisi de requisits ]

A l'últim pas de l'apartat *anàlisi de requisits* s'ha cercat el programari necessari per a la realització del projecte. En aquesta tasca caldrà instal·lar aquest programari i fer-lo funcionar:

- Eines ofimàtiques (MS Word 2010, MS Project 2013).
- Eina de dibuix UML (Magic Draw v.17 PE).
- El SGBD.

### 2.2.1.3 Definició de la estratègia del TFG1

[ Duració: 0.75 dies - Predecessora: Instal·lació del programari ]

---

<sup>1</sup> En el moment en què s'ha realitzat aquesta documentació ja s'havia finalitzat aquesta tasca. La definició de la metodologia de treball escollida es pot veure a l'apartat 2 d'aquest document.

Aquesta tasca fa referència a la definició de la estratègia a prendre per a organitzar la realització del treball de fi de grau. Això inclou:

- Escollir un mètode de treball (cascada, àgil,...).
- Definir les tasques principals que conformaran el TFG.
- Definir les tasques específiques que conformen cadascuna de les tasques principals.

### 2.2.1.4 Creació d'un diagrama de Gantt

[ Duració: 0.75 dies - Predecessora: Definició de la estratègia del TFG]

Un cop definides les tasques que conformen el projecte, es realitza un diagrama de Gantt mitjançant el MS Project per a representar visualment la planificació.

A continuació es mostren els diagrames corresponents a la planificació de totes les fases del projecte (es pot consultar l'arxiu en format MS Project [Pla de treball.mpp](#) als annexes per a una informació més detallada) :

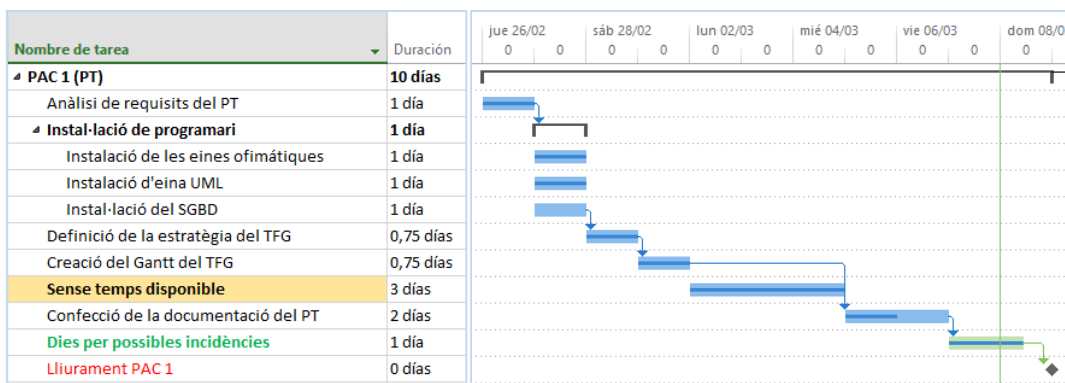


Fig. 2.2 Diagrama Gantt de la PAC 1

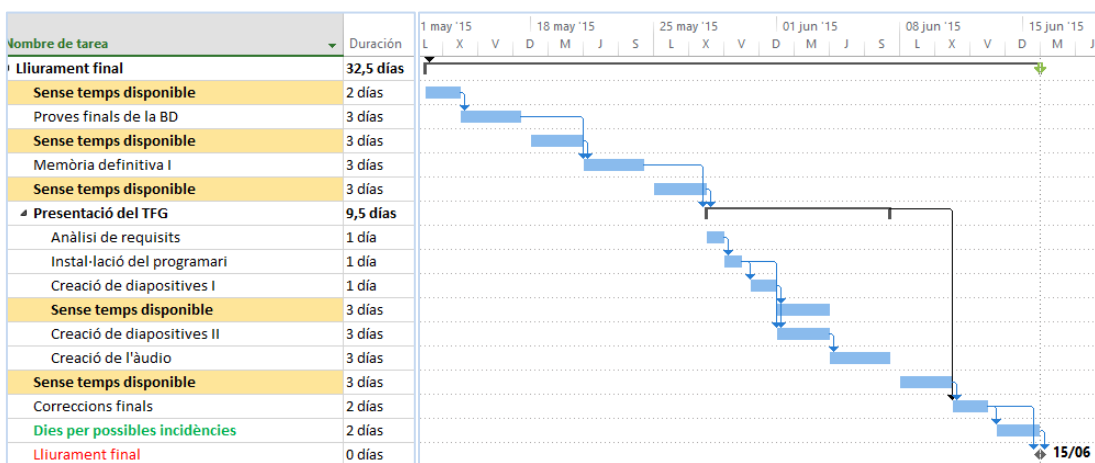


Fig. 2.3 Diagrama Gantt del lliurament final

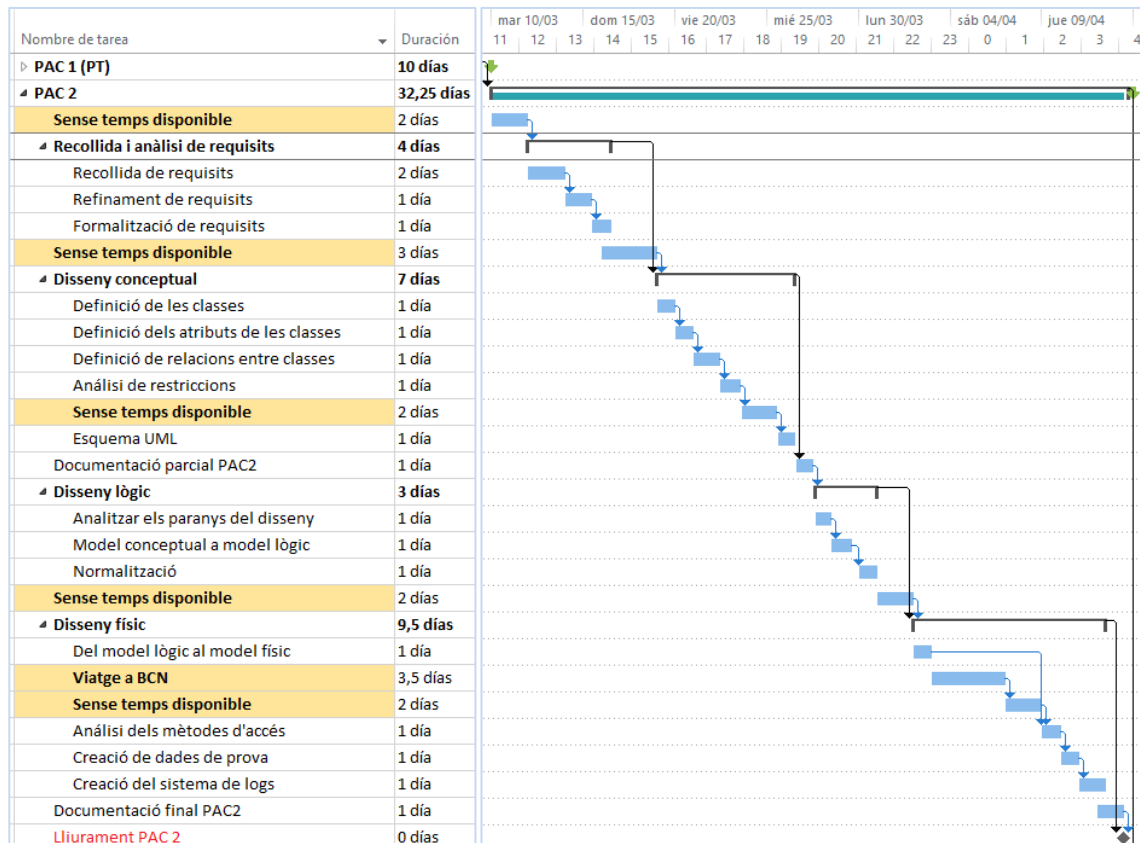


Fig. 2.4 Diagrama Gantt de la PAC 2

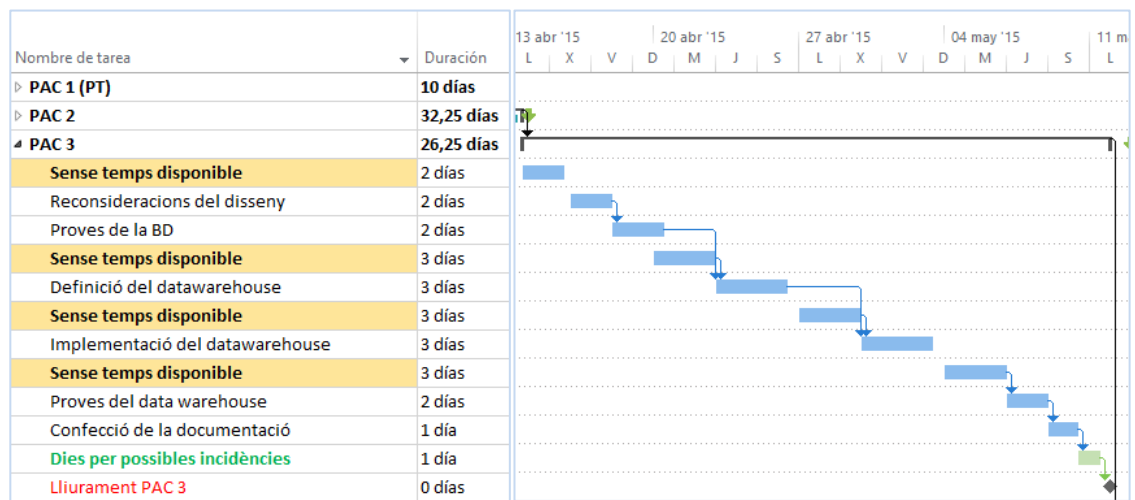


Fig. 2.5 Diagrama Gantt de la PAC 3

### 2.2.1.5 Confecció de la documentació del pla de treball

[ Duració: 2 dies - Predecessora: Creació del diagrama Gantt ]

La última tasca d'aquest bloc del pla de treball abans del lliurament és la confecció del propi pla de treball. Conté tots els apartats especificats a la documentació de l'assignatura<sup>2</sup>, que són els següents:

- Objectius del treball.
- Metodologia de treball.
- Tasques en què es dividirà el projecte.
- Fites i temporització d'aquestes.
- Possibles eventualitats.
- Diagrama Gantt.
- Material necessari.

#### **2.2.1.6 Possibles incidències**

[ Duració: 1 dies - Predecessora: Creació de la documentació del PT ]

Es reserva un dia com a mínim dins d'aquest bloc per a fer front a possibles incidències imprevistes que puguin sorgir. Cada setmana té 3 dies bloquejats per tasques alienes al projecte, però es pot donar el cas que un d'aquests dies sigui disponible per a treballar en el projecte. Es per això que no es creu oportú reservar més dies per a incidències.

#### **2.2.1.7 Lliurament del pla de treball**

[ Predecessora: Possibles incidències ]

Un cop confeccionada tota la documentació referent al pla de treball, es lliurará a la bústia de l'assignatura abans de la data límit.

### **2.2.2 PAC 2**

[ Interval 10 de març –13 d'abril ]

El segon bloc és el que presenta més activitat ja que és on es realitza l'estructura principal del projecte. Per a realitzar un plantejament coherent es faran servir els apunts de l'assignatura *Disseny de bases de dades* com a guia.

#### **2.2.2.1 Recollida i anàlisi de requisits**

[ Duració: 4 dies]

El primer que es farà abans d'iniciar la fase del disseny serà analitzar el més detalladament possible tots els requisits necessaris que presenta l'enunciat del projecte. Això implica les següents fases:

---

<sup>2</sup> Concretament els apartats descrits a l'apartat Pla de treball del document "Recomanacions TFG".

- Recollida de requisits: analitzar quin tipus d'usuari farà servir la base de dades, estudiar l'entorn i l'ús que es farà de l'aplicació i resoldre els dubtes al fòrum de l'assignatura.
- Refinament de requisits: mantenir contacte constant amb el fòrum per a detallar cada requisit nou que sorgeixi.
- Formalització de requisits: convertir els requisits a un format estructurat.

### **2.2.2.2 Disseny conceptual**

[ Duració: 5 dies - Predecessora: Recollida i anàlisi de requisits ]

El disseny conceptual reflexa de manera gràfica les característiques de la base de dades extretes de l'enunciat. Consta del següent procés:

- Definició de les classes.
- Definició dels atributs de les classes.
- Definició de relacions entre classes (n-àries, reflexives, d'herència, d'agregació,...).
- Anàlisi de restriccions per tal de mantenir la integritat a la base de dades.
- Esquema UML: fent servir la eina Magic Draw caldrà representar tot el disseny conceptual.

### **2.2.2.3 Documentació parcial de la PAC2**

Arribat aquest punt força informació rellevant estarà disponible, i es oportú aturar-se un dia i recopilar-la de cara a la entrega final. També s'actualitzarà l'esborrany de memòria si la disponibilitat temporal és suficient.

### **2.2.2.4 Disseny lògic**

[ Duració: 3 dies - Predecessora: Disseny conceptual ]

A aquesta fase de disseny s'interpretarà el resultat del disseny conceptual per tal d'adaptar al model relacional de bases de dades al qual pertany el sistema Oracle fet servir a aquest projecte. Això comporta:

- Analitzar els paranys del disseny que pot contenir el disseny conceptual previ.
- Transformar el disseny conceptual al model lògic.
- Aplicar les regles de normalització per a comprovar la integritat del plantejament.

### **2.2.2.5 Disseny físic**

[ Duració: 5 dies - Predecessora: Disseny lògic ]

L'últim apartat del disseny tracta d'implementar tot el disseny definit fins al moment al SGBD. S'ha estructurat aquesta part de la següent manera:

- Fer la conversió del model lògic al model físic (creació d'usuaris i taules).
- Anàlisi dels mètodes d'accés (índex, seqüències i disparadors).

- Creació del conjunt de dades de prova i creació del sistema de *logs*. Aquestes dues tasques són independents de la resta i es poden realitzar simultàniament.
- Processament de consultes i vistes.

El temps és força limitat per a una tasca d'aquest tipus, tot i que s'intentarà dur a terme en el temps establert. Si calgués dedicar-hi més temps, arribat el moment s'adaptarà el pla de treball per a fer lloc a aquesta tasca. Els dies disponibles s'hauran d'escurçar a la PAC3, tot i que el temps disponible es força ampli i no hauria de ser complicat trobar aquest espai. Es per aquest motiu que es prescindeix de dies per possibles incidències en aquest bloc.

#### **2.2.2.6 Confecció de la documentació**

[ Duració: tasca continuada ]

A mesura que es va avançant en el projecte s'anirà documentant tot per tal d'anar confeccionant la memòria final. Abans de començar però, caldrà analitzar quin format ha de tenir el document per tal de complir amb la normativa de l'assignatura.

Aquesta tasca s'ha considerat independent de la resta ja que es pot realitzar al llarg de tota la fase un cop determinat el format del document.

#### **2.2.2.7 Lliurament de la pac 2**

[ Predecessores: Disseny físic i confecció de la documentació ]

L'última tasca d'aquest bloc serà entregar la documentació sobre el desenvolupament del projecte fet durant aquest temps.

### **2.2.3 PAC 3**

[ Interval 13 d'abril - 11 de maig ]

La pac 3 es destinarà a realitzar canvis sobre el disseny inicial en base a possibles errors o mancances que s'hagin pogut trobar realitzant la segona pac, el que comportarà una iteració completa sobre el procés ja realitzat. També es durà el desenvolupament del sistema de *data warehouse* descrit a l'enunciat.

#### **2.2.3.1 Reconsideracions del disseny**

[ Duració: 2 dies ]

Es poc probable que en aquest punt del projecte sigui necessari realitzar canvis en la estructura del disseny, però per si de cas s'han reservat un parell de dies a aquest objectiu a l'inici de la tercera pac.

### **2.2.3.2 Proves sobre la base de dades**

[ Duració: 2 dies - Predecessora: Reconsideracions del disseny ]

Aquesta tasca tracta la creació d'un conjunt suficient de dades per a carregar la base de dades. Sobre aquestes es realitzarà un minuciós joc de proves per tal de descobrir possibles errors en la integritat del sistema.

### **2.2.3.3 Creació del data warehouse**

[ Duració: 8 dies - Predecessora: Proves de la base de dades ]

L'últim punt que es tractarà un cop desenvolupada la base de dades serà el del *data warehouse*. Es un tema desconegut que caldrà estudiar a fons, motiu pel qual es reserva una quantitat de temps suficient. Com en el cas de l'apartat 3.3.1 sobre reconsideracions del disseny, seria bo poder enllestir abans de temps la tasca per a no patir mancances de temps per imprevistos, tot i que ja s'han reservat uns dies per aquest motiu.

Aquest apartat constarà de 3 fases:

- Definició
- Implementació
- Proves

### **2.2.3.4 Proves sobre el data warehouse**

[ Duració: 2 dies - Predecessora: Creació del *data warehouse* ]

En aquest moment no es possible definir detalladament quin tipus de proves caldrà realitzar sobre aquest tema però s'actualitzarà aquest apartat un cop estudiat el tema a l'apartat anterior.

### **2.2.3.5 Confecció de la documentació**

[ Duració: tasca continuada ]

D'igual manera que es va fer durant la realització de la segona pac, la documentació d'aquest apartat s'anirà realitzant simultàniament a la realització de les tasques d'aquest tercer bloc. El format del document ja s'ha definit anteriorment, així que només caldrà continuar la memòria.

### **2.2.3.6 Lliurament de la pac 3**

[ Predecessores: Possibles incidències i confecció de la documentació ]

Un cop realitzades totes les tasques i confeccionada la documentació, caldrà lliurar la feina feta fins al moment abans de la data límit.

## **2.2.4 Lliurament final**

[ Interval 12 de maig – 15 de juny ]

Aquest bloc correspon a la part final del projecte i en ell caldrà finalitzar la memòria, realitzar les últimes proves de la implementació de la base de dades i crear la presentació del projecte.

### **2.2.4.1 Proves finals sobre la base de dades**

[ Duració: 3 dies ]

Es realitzaran les últimes proves sobre la base de dades de cara a corregir errors d'última hora que hagin pogut sorgir

### **2.2.4.2 Memòria definitiva**

[ Duració: 3 dies - Predecessora: Proves finals sobre la BD ]

Caldrà finalitzar la memòria del projecte amb les últimes dades del desenvolupament.

### **2.2.4.3 Presentació del projecte**

[ Duració: 9 dies - Predecessora: Memòria definitiva ]

Un cop finalitzada la memòria i la implementació de la base de dades, serà el moment de crear la presentació que acompanyarà al projecte. Per tal de realitzar aquesta tasca de manera coherent, serà necessari:

- Analitzar els requisits necessaris per a realitzar-la llegint la normativa referent a aquest tema i preparant el programari necessari per la creació de la presentació.
- Instal·lació del programari de diapositives i d'àudio i vídeo.
- Creació de les diapositives.
- Gravació de la pista d'àudio.

### **2.2.4.4 Últimes correccions**

[ Duració: 2 dies - Predecessora: Presentació del TFG ]

Apartat destinat a realitzar les darreres modificacions i els retocs finals de la documentació del projecte.

### **2.2.4.5 Possibles incidències**

[ Duració: 2 dies ]



#### **2.2.4.6 Lliurament final**

[ Predecessora: Últimes correccions ]

Lliurament de la memòria final i de la presentació del treball de fi de grau.

### **2.3 Gestió del temps**

#### **2.3.1 PAC 1**

L'inici del projecte ha transcorregut sense cap incidència remarcable. S'ha realitzat la confecció del pla de treball segons els requeriments i s'ha entregat a temps.

Després d'analitzar els requisits del pla de treball s'han instal·lat les següents eines ofimàtiques per a la realització del treball:

- SGBD MySQL versió 5.6.21. Comentar breument que la eina per defecte al llarg dels estudis a estat Oracle Standard, però vaig contactar amb el tutor per poder fer servir MySQL per a aquest projecte ja que actualment és la meua eina de treball a la feina i suposa un aprenentatge molt útil per mi.
- MS Project 2010 per a la realització del pla de treball
- MS Word 2010 per a la redacció de la memòria.
- HeidiSQL 9.2 com a client MySQL.
- DB Forge Studio MySQL, client MySQL que compte amb un *debugger* de procediments emmagatzemats.
- MS Visio 2010 per a realitzar els diagrames UML.

#### **2.3.2 PAC 2**

Aquesta etapa del projecte a transcorregut també sense grans incidències, a part de les previstes al pla de treball inicial. Ha estat , però, força complicat obtenir una versió definitiva del diagrama UML per tal de representar els requisits de l'enunciat. De fet, després d'haver realitzat la implementació del model físic he hagut de realitzar diverses iteracions per a modificar el disseny lògic ja que hi havia diverses parts que no complien els requeriments. Aquesta ha estat una part complicada de cara a complir amb els terminis marcats al pla de treball, ja que s'ha perdut molt de temps que amb un bon disseny inicial hauria estalviat.

El sistema de logs i els procediments ABM no han estat implementats. La manca de temps degut a les diferents correccions realitzades així com una mala aproximació del temps necessari per a cada tasca han fet que aquests dos apartats s'hagin de realitzar amb el temps disponible per a la pac 3. Per tal de poder complir amb els terminis, s'han fet servir els dies disponibles per a incidències a la pac 3 així com temps destinat per a crear el joc de dades i les proves de la base de dades.

També és una mancança important el fet que el magatzem de dades encara no s'ha considerat i això pot provocar futurs canvis en el disseny en un moment amb que no es compte amb marge per a incidències.

### **2.3.3 PAC 3**

Aquesta etapa ha estat marcada per un problema de salut pel qual ha estat impossible treballar en el projecte durant uns dies juntament amb unes setmanes intensives a la feina . Afortunadament he pogut compensar una mica amb dies que pensava, en un principi, que estaria ocupat. Com a punt positiu, la salut ha tornat i el coll d'ampolla a la feina s'ha reduït, el que em fa pensar que tot el temps perdut es podrà recuperar a l'etapa final

També el meu desconeixement sobre els magatzems de dades ha fet que una gran part del temps disponible l'hagués de fer servir per a cercar informació, el que ha deixat encara menys temps disponible per a la implementació del mòdul estadístic. Finalment he pogut fer un primer esbós del magatzem, implementar una primera taula de fets per a quatre de les estadístiques requerides i provar-ho amb èxit. Cal dir que he hagut de modificar diverses part del model lògic inicial degut a aquesta part, ja que implicava tenir una sèrie de relacions que sense el magatzem semblaven inicialment correctes. Per les proves he hagut de crear alguns procediments per tal de generar dades de prova, el que ha acabat consumint el poc temps que quedava a aquesta pac 3. Considero que, tot i que encara queda feina a fer amb el magatzem de dades, la recerca inicial ha finalitzat i ara puc dedicar el temps a implementar la solució.

La part positiva es que he pogut realitzar satisfactòriament els següents apartats:

- Procediments d'ABM.
- Tractaments d'errors i taula de log.
- Disparadors de prova per a inserció de dades poc lògiques.
- Primer disseny i implementació del magatzem de dades.
- Joc de proves per a aquests apartats.
- Confecció de la memòria relativa a aquestes parts, tot i que manca el referent al magatzem de dades.

De cara al lliurament final he reservat 3 dies per a implementar la resta del magatzem de dades i he mogut les tasques previstes respectivament. El fet que la memòria va força bé i que comptaré amb algun dia lliure addicional em fa pensar que el projecte arribarà a bon port.

### **2.3.4 Lliurament final**

Aquesta etapa del projecte s'ha vist molt condicionada per la implementació del magatzem de dades. Diversos dissenys i diversos contactes amb el tutor han estat necessaris fins a obtenir la solució idònia. El problema principal ha estat un error de concepte: inicialment pensava que la base de dades era plena de dades quan s'havia d'omplir el magatzem, mentre que el que es demanava era omplir-lo incrementalment. Els procediments implementats per al primer disseny,

però, formen part d'una funcionalitat addicional incorporada al projecte anomenat estadístiques absolutes (veure apartat [6.2.2](#)).

Aquesta "pèrdua de temps" (que comptabilitzo en uns 10 dies perduts) ha fet que el pla inicial quedés obsolet, ja que només es comptava amb uns pocs dies d'incidències. Així que algunes tasques s'han vist afectades o, fins i tot, han hagut de deixar-se sense realitzar. És el cas de les modificacions dels procediments de modificació per tal d'actualitzar les estadístiques, el qual no ha estat implementat. També l'etapa de test no ha estat tan exhaustiva com caldria. Tot i que els procediments i les estadístiques funcionen correctament, les proves creuades de dades errònies i de robustesa del magatzem de dades no s'ha dut a terme de manera concisa. La resta de temps necessari s'ha extret d'altres tasques que han vist reduït el nombre d'hores, com per exemple la presentació (inicialment es volia fer un *screencast* però ha estat impossible), repàs de la memòria o refinament dels arxius de proves.

El resultat, però, es satisfactori i crec que la solució compleix amb tots els requeriments especificats a l'enunciat.

## 3. Disseny conceptual

Aquesta és la primera fase del disseny d'aquest sistema de base de dades i tracta de consolidar els requeriments definits pel client. Els objectius d'aquesta fase són:

- Analitzar detalladament els requisits.
- Definir els atributs de cada entitat escollint un tipus de dades adient.
- Crear un diagrama UML que presenti les relacions entre les entitats necessàries per a representar els requeriments del sistema.
- Definir la multiplicitat, les restriccions i el tipus de dependència entre les entitats reflectint els requeriments del sistema.

### 3.1 Anàlisi de requisits

Els sistema presenta els següents requisits bàsics:

- D'igual manera que la Fórmula 1, les altres categories dependents de la FIA han d'estar registrades al sistema. El WRC i la Formula 3 en són un exemple.
- Les entitats que participen en les competicions automobilístiques han d'estar registrades al sistema. Entitats bàsiques amb què ha de comptar la base de dades són equips, pilots, circuits, fabricants, components, cotxes, patrocinadors, carreres, competicions, resultats i telemetria.
- S'ha d'enregistrar els resultat dels pilots a cada cursa.
- Cal considerar que un pilot pot conduir qualsevol cotxe d'un equip durant una competició, però que no pot canviar d'equip en un mateix any.
- Cada equip, en un mateix any, només podrà tenir un màxim de 2 pilots.

A més, per tal de fer més realista el model, s'han afegit les següents requisits:

- Cada patrocinador pertanyerà a un sector concret, com per exemple l'energètic, la banca o el de l'automoció.
- Cada equip tindrà un director d'equip.
- Es crea l'entitat *Circuit* per tal de desar informació addicional i millorar la independència entre entitats.
- Es crea l'entitat *Emissora* per representar un medi audiovisual amb drets d'emissió sobre una categoria concreta. Aquesta entitat s'afegeix per considerar-la suficientment important al món del motor.

## 3.2 Definició detallada d'entitats i atributs.

### 3.2.1 Competició

Representa cadascuna de les categories de la FIA (per exemple F1, WRC o CIK). El nom o les sigles serviren de clau primària, però per conservar el format general de la base de dades es crearà una columna amb l'identificador: Els atributs de l'entitat són els següents:

- **codi\_competició** (PK): Identificador únic de la categoria.
- **nom**: nom de la competició.
- **acrònim**: sigles de la competició.
- **descripció**: descripció de la competició.
- **temporada**: any en que es celebra la competició.

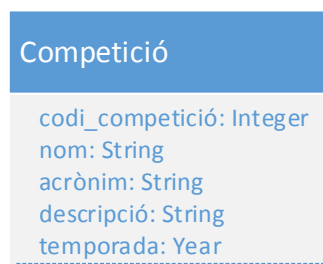


Fig. 3.1 Entitat Competició en format UML

### 3.2.2 Sector

Per tal d'establir una llista de sectors industrials fixa, s'ha creat una classe específica amb una sèrie de sectors bàsics definits que poden ser modificats segons les necessitats en un futur. Els atributs de l'entitat són els següents:

- **codi\_sector** (PK): identificador únic del sector.
- **nom**: nom descriptiu del sector empresarial.



Fig. 3.2 Entitat Sector en format UML

### 3.2.3 País

Hi ha diverses entitats que contenen com a atribut la seva nacionalitat. Desar aquesta informació per separat a cada entitat pot produir errors per sintaxi o d'escriptura. Es per això que s'ha creat aquesta taula per tal d'unificar la nomenclatura dels diferents països. Els atributs de l'entitat són els següents:

- **codi\_país** (PK): sigles internacionals dels països.
- **nom**: nom del país.



Fig. 3.3 Entitat País en format UML

### 3.2.4 Persona

Superclasse que representa una persona física. Als requeriments es presenta la necessitat de representar el manager de cada equip així com els pilots, però hi cap la possibilitat de que altres càrrecs siguin necessaris. Així que, per evitar duplicitats, aquesta classe desarà tota la informació bàsica de cada persona i els atributs específics de cada persona es desaran a les subclasses.

- **codi\_persona**: codi d'identificació de la persona.
- **nom**: nom real de la persona.

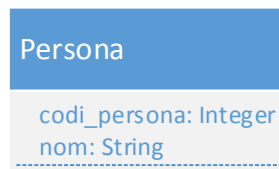


Fig. 3.4 Entitat Persona en format UML

### 3.2.5 Equip

Representa els equips enregistrats al sistema. Els atributs de l'entitat són els següents:

- **codi\_equip** (PK): identificador únic de l'equip.
- **nom**: nom de l'equip.
- **seu**: ciutat on l'equip té la seu oficial.



Fig. 3.5 Entitat Equip en format UML

### 3.2.6 Director

Subclasse de Persona, representa el director d'un equip. Als requisits es proposa el nom "manager" però s'ha cregut convenient anomenar la classe director per fer servir el mot en català com a la resta de classes i per ser el nom del càrrec real (director d'equip). Es desaran

els estudis del director perquè es un detall que es troba molt sovint a les fitxes públiques dels directors. Els atributs de l'entitat són els següents:

- **estudis**: estudis o professió del director.



Fig. 3.6 Entitat Director en format UML

### 3.2.7 Cotxe

Representa cadascun dels cotxes registrats al sistema. Els atributs de l'entitat són els següents:

- **codi\_cotxe** (PK): identificador únic del cotxe.
- **model**: cadena per nomenar el model del cotxe.
- **any\_debut**: atribut per desar l'any de la primera carrera.

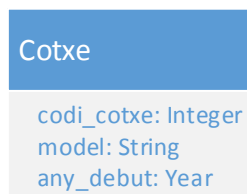


Fig. 3.7 Entitat Cotxe en format UML

### 3.2.8 Patrocinador

Representa els patrocinadors enregistrats al sistema. Els atributs de l'entitat són els següents:

- **codi\_patrocinador** (PK): identificador únic del patrocinador.
- **nom**: nom del patrocinador
- **sector**: àmbit de negoci del patrocinador. Els atributs de l'entitat són els següents:

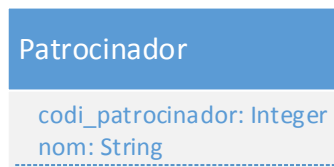


Fig. 3.8 Entitat Patrocinador en format UML

### 3.2.9 Fabricant

Representa els fabricants de components enregistrats al sistema. Els atributs de l'entitat són els següents:

- **codi\_fabricant** (PK): identificador únic del fabricant.

- **nom**: nom del fabricant
- **any\_debut**: atribut per desar la data d'entrada a la competició.



Fig. 3.9 Entitat Fabricant en format UML

### 3.2.10 Component

Representa els components enregistrats al sistema fets servir pels cotxes. Els atributs de l'entitat són els següents:

- **codi\_component** (PK): identificador únic del component.
- **descripció**: descripció del component.
- **utilitat**: descripció abreujada de per què es farà servir el component.



Fig. 3.10 Entitat Component en format UML

### 3.2.11 Circuit

Representa cadascun dels circuits disponibles al sistema. Als requeriments la informació relativa al circuit es troba dins de la classe 'carrera', però s'ha decidit introduir la classe circuit per a poder desar més informació relativa sobre aquesta entitat, representar més fidelment la realitat (un circuit és una entitat amb pes a totes les competicions) així com poder obtenir informació més endavant directament de la taula en qüestió. Els atributs de l'entitat són els següents:

- **codi\_circuit** (PK): identificador únic del circuit.
- **nom**: nom del circuit.
- **longitud**: longitud en metres del circuit.



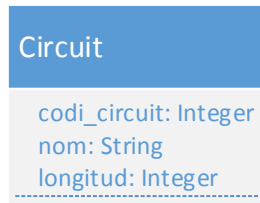


Fig. 3.11 Entitat Circuit en format UML

### 3.2.12 Emissora

Representa una emissora de televisió. S'ha afegit aquesta classe extra a l'enunciat per tal de representar més fidelment la realitat (les emissores són essencials al món de l'automobilisme ja que són la base dels beneficis; pels diners que aporten i per mostrar els patrocinadors al gran públic). Els atributs de l'entitat són els següents:

- **codi\_emissora** (PK): identificador únic de l'emissora.
- **nom**: nom de l'emissora.
- **nacionalitat**: país en el qual emetrà l'emissora.



Fig. 3.12 Entitat Emissora en format UML

### 3.2.13 Pilot

Subclasse de *Persona*, representa els pilots enregistrats al sistema. Els atributs de l'entitat són els següents:

- **codi\_llicència**: Codi de llicència del pilot.



Fig. 3.13 Entitat Pilot en format UML

### 3.2.14 Telemetria

Cada fila d'aquesta taula representa una dada telemètrica enviada per un component en concret. Els atributs de l'entitat són els següents:

- **codi\_dada** (PK): Identificador únic de la dada telemètrica.
- **moment**: data fins a la mil·lèsima quan es va rebre la dada.
- **àmbit**: Tipus de dada (temperatura, consum, marxa, velocitat,...)
- **valor**: Valor numèric de la mesura.

- **unitat:** Unitat de la mesura (segons, litres, cap,...).

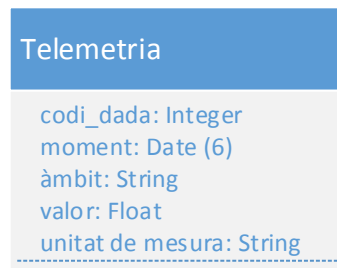


Fig. 3.14 Entitat Telemetria en format UML

### 3.3 Relacions entre les entitats

En aquest apartat s’analitzen les peculiaritats de cada relació amb les demés entitats, raonant el motiu pel qual s’han prés finalment les decisions presents al disseny UM (Es pot consultar el diagrama complet al final de l’apartat).

#### 3.3.1 L’entitat Equip

Aquesta entitat té múltiples relacions que inclouen *Patrocinador*, *Pilot* i *Director*. Els patrocinadors, el quals realitzen una aportació econòmica als equips, es relacionen amb els equips mitjançant un *Patrocini*. Al no haver-hi cap restricció als requeriments inicials, s’ha considerat que un patrocinador pot patrocinar diversos equips . També s’han afegit dos atributs a la relació que poden ser útils i la fan més realista: el *cost* d’aquest patrocini i seva duració en temporades (s’ha considerat que un patrocinador no pot ser-ho per menys d’una temporada). D’aquesta manera es podrà saber si un contracte és vigent o no i desar un historial.

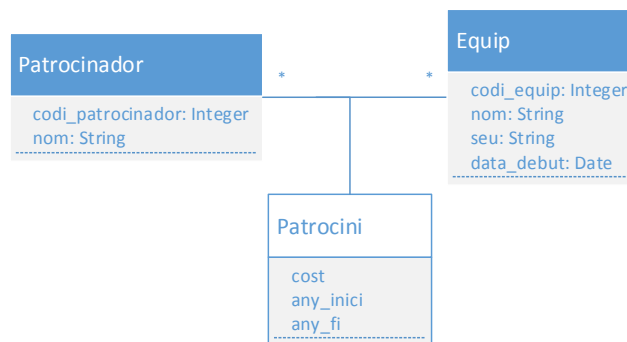


Fig. 3.15 Relació Patrocini entre les entitats Equip i Patrocinador.

Els equips poden tenir com a màxim dos pilots, es per això que s’ha creat una relació amb *Pilot* amb una multiplicitat 1 a 2. A la vegada, un pilot estableix una relació contractual amb un equip per, com a mínim, una temporada. Aquesta informació es desarà a l’entitat associativa *Contractat* com a any d’inici i any de final de contracte.

També es considera que un *Equip* tindrà sempre un *Director*, al mateix temps que un *Director* només es podrà fer responsable d’un *Equip*. S’ha creat una classe associativa entre aquestes

entitats per tal de desar, a més, la informació relativa a la duració temporal entre els dos. Aquesta duració es mesura en anys per tal de representar que els contractes entre equips i directors es mesuren per temporades.

La relació entre pilot i equip es connecta amb l'entitat temporada per tal de desar un històric de la relació necessari per al magatzem de dades.

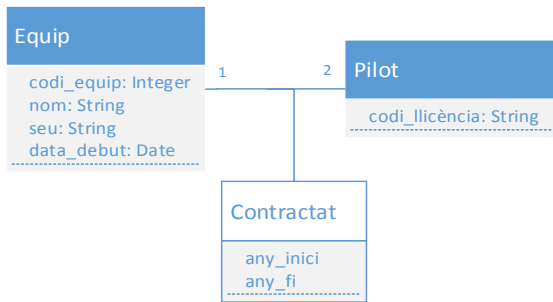


Fig. 3.16 Relació Contractat entre Equip i Pilot.

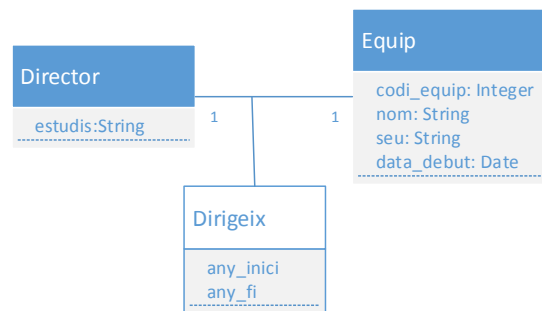


Fig. 3.17 Relació Patrocini entre Equip i Director.

Per últim *Equip* es relacionarà amb la entitat *Cotxe* ja que cada equip en té diversos. Però un cotxe no ha de ser exclusivament d'un equip, ja que a algunes competicions com el WRC diversos equips fan servir el mateix cotxe.

La relació es connecta amb l'entitat temporada per tal de desar un històric de la relació necessari per al magatzem de dades.

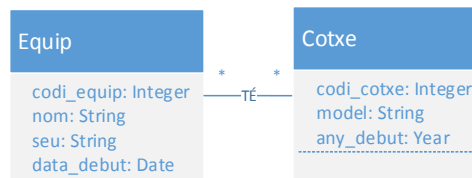


Fig. 3.18 Relació Té entre les entitats Equip i Cotxe.

### 3.3.2 L'entitat Cursa

Aquesta és una entitat amb molta importància, ja que porta gran part del pes significatiu del disseny. Representa les curses anuals de cada competició, tenint en compte les següent relacions:

- Un **pilot**, una **competició** i un **circuit** concret pot donar-se a diverses **dates** (multiplicitat \* a dates).
- Una **competició**, a un **circuit** i una **data** concretes compte amb diversos **pilots** (multiplicitat \* a pilots).
- Un **circuit**, una **data** i un **pilot** concrets només poden pertànyer a una **competició** (multiplicitat 1 a competició).

- Una **data**, un **pilot** i una **competició** concreta només poden donar-se a un **circuit** (multiplicitat 1 a circuit).

Tot i que la seva importància recau en què és l'entitat associativa entre Pilot i Circuit i que desarà tots els resultats de cada carrera. D'aquesta manera es representa el fet que una cursa és una concentració de pilots que competeixen a un mateix circuit a una data concreta, relació de la qual sorgeixen uns resultats en forma de posició final de cada pilot (amb aquests resultats es podrà establir una puntuació per pilot, que és un dels requeriments del disseny).

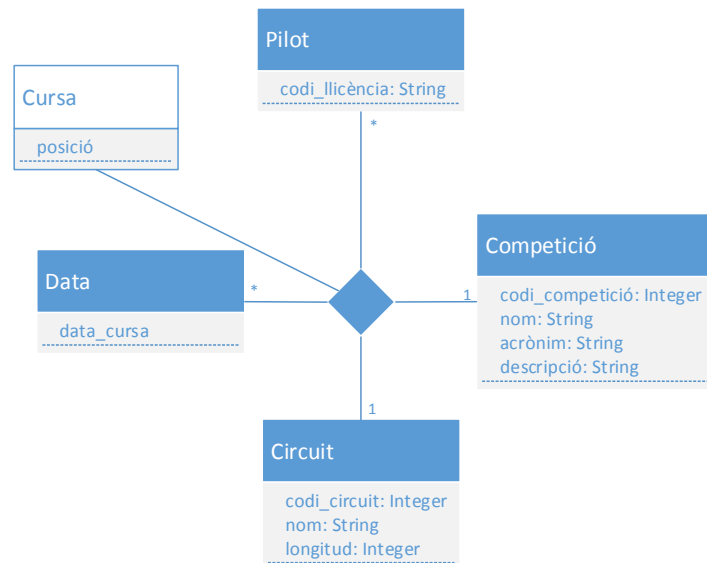


Fig. 3.19 Relació entre Circuit, Pilot i Competició al voltant de Cursa.

### 3.3.3 L'entitat Component

Representa una de les peces del cotxe proveïdes per un fabricant concret, de les quals es vol desar una descripció i la seva utilitat un cop instal·lat al cotxe. Com es veu al diagrama, s'ha considerat que un cotxe pot tenir múltiples components però que un component és únic i només pot ser utilitzat per un cotxe. A més cada component es considera que prové d'un sol fabricant, relació anomenada *Proporciona*.

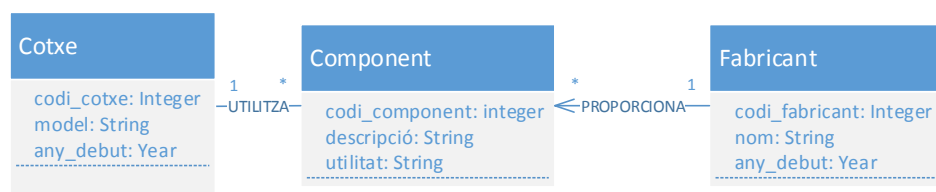


Fig. 3.20 Relacions Utilitza entre Cotxe i Component, i Proporciona entre Fabricant i Component.

L'altre relació en què *Component* està implicada és amb *Telemetria*. Això és degut a que cada dada telemètrica provindrà sempre d'un component concret o, dit d'una altra manera, cada component *Genera* una dada de telemetria.

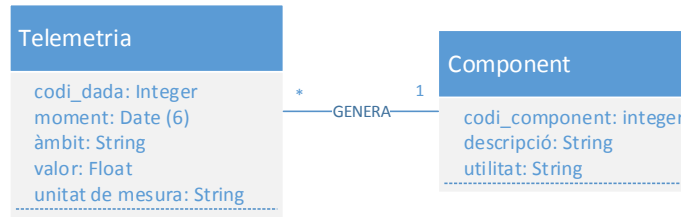


Fig. 3.21 Relació Genera entre Telemetria i Component.

### 3.3.4 L'entitat Persona

Al nostre sistema hi haurà dos tipus de persones implicades (representades per les entitats *Director* i *Persona*) però és possible que en un futur hi hagin més. Per tal de generalitzar els atributs bàsics de cada persona s'ha creat aquesta entitat. Es considera que tants els directors com els pilots tenen feines dedicades i, per tant, aquesta generalització mai serà encavalcada. El ventall de persones, però, pot ser molt més ampli, així que la generalització serà parcial.

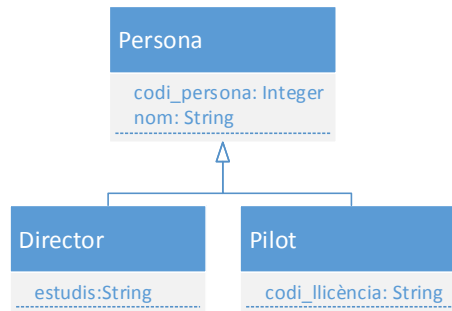


Fig. 3.22 Relació de generalització entre Persona, Director i Pilot.

### 3.3.5 L'entitat Retransmissió

Aquesta entitat associativa serveix per desar quines emissores tenen drets audiovisuals sobre una competició en concret. Hi ha diverses emissores per a cada competició i cada emissora tindrà drets d'emissió per a diverses competicions.



Fig. 3.23: Relació Retransmissió entre les entitats Emissora i Competició.

### 3.3.6 L'entitat País

Aquesta entitat té la funció d'estandaritzar la manera en que l'atribut referent al país apareix a les entitats. *Persona* té una nacionalitat, *Circuit* es troba a un país i *Emissora* té drets

audiovisuals a uns països concrets. Totes tres referencien *País*, la qual té com a clau primària el codi ISO<sup>3</sup> de cada país.

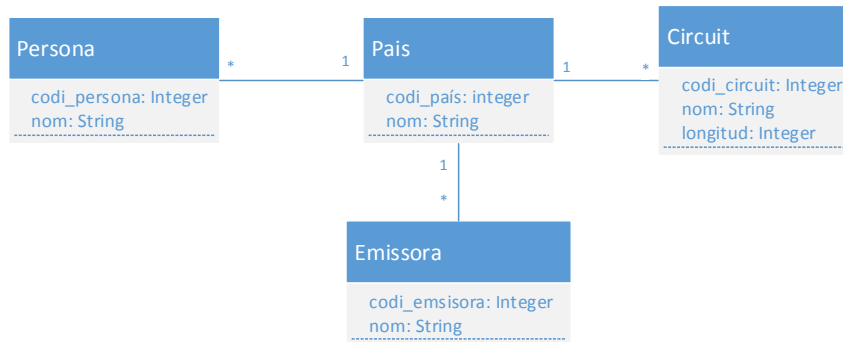


Fig. 3.24: Relació entre les entitats Persona, Circuit i Emissora vers País

### 3.4 Diagrama UML

A continuació es pot consultar el diagrama final amb totes les relacions anteriorment descrites. Aquest diagrama és el resultat de petites però diverses modificacions al llarg del projecte per adaptar-lo a totes les necessitats de l'enunciat (als arxius annexes es pot trobar el diagrama relacional UML sota el nom Diagrames UML en format PDF y VSDX per a una visió més detallada).

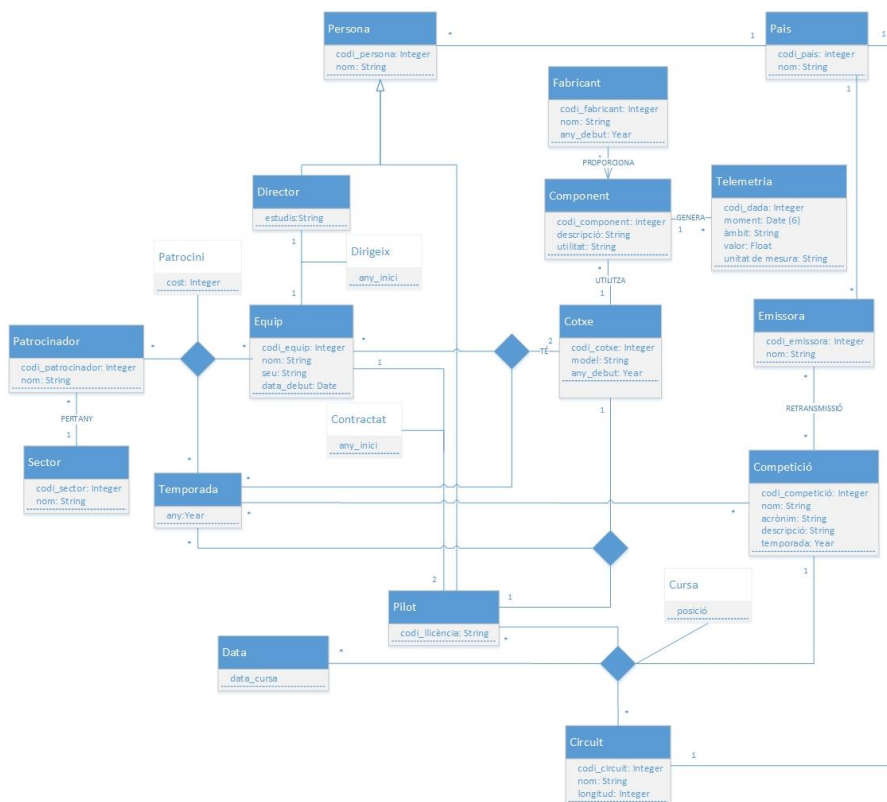


Fig. 3.25 Diagrama UML complet

<sup>3</sup> Concretament s'ha escollit la norma ISO 31366-1 alpha 2 basada en la norma 31366:1993 que defineix cada país mitjançant dues lletres.

## 4. Disseny lògic

### 4.1 Transformació del model conceptual al model relacional

L'entitat *Equip* es relaciona amb diverses entitats:

- La relació que mantenen les dues entitats *Patrocinador* i *Equip* s'anomenarà "Patrocini". Com un equip té molts patrocinadors, la multiplicitat entre ells serà molts a molts. Per tant, es crearà una nova classe (*Patrocini*) per a representar aquesta relació de multiplicitat.
- La persona que dirigeix l'equip serà representada per l'entitat *Director* amb multiplicitat 1-1. La relació associativa entre elles anomenada *Dirigeix* es representarà afegint una clau forana de *Equip* a *Director* així com els atributs de la relació associativa.
- Cada equip tindrà un màxim de dos pilots així que cal una relació amb l'entitat *Pilot* de multiplicitat 1 a molts representada per una clau forana a la taula *Pilot* (per ser aquella entitat amb la multiplicitat major que 1).
- Per representar la relació molts a molts entre *Cotxe* i *Equip* es crea una nova entitat *Cotxe\_Equip* amb clau primària la combinació de les claus de les dues entitats.

La multiplicitat entre *Sector* i *Patrocinador* és 1 a molts, així que s'afegirà una clau forana al costat amb multiplicitat múltiple.

Per generalitzar la representació d'una persona s'ha creat l'entitat del mateix nom. En el nostre cas només tindrà dues subclasses (director o pilot) les quals no es poden encavalcar mai. Per tant, per tal de representar aquesta relació i evitar atributs nulls i repeticions, es crearà una nova entitat per a cada subclasse i per la classe, relacionades mitjançant claus foranes.

*Cursa* és una entitat associativa que sorgeix entre pilot i circuit. Per a representar això i tenint en compte que la multiplicitat es molts a molts, es crea una entitat independent (*Cursa*) amb les claus foranes de *Pilot* i *Circuit* com a clau primària.

Emissora té una relació amb multiplicitat molts a molts amb cursa i, per tant, es genera una nova entitat associativa amb clau primària la combinació de les claus de les entitats de l'associació.

*Farbricant* i *Component* són dues entitats amb una relació de 1 a molts. Cada component és únic i pertany a només un fabricant, a part de que un mateix tipus de component pugui ser fabricat per més d'un fabricant. Component i cotxe també tenen una relació 1 a molts ja que un component específic que pertany a un cotxe no pot, a la vegada, pertànyer a una altre cotxe. Aquestes dues relacions quedaran representades amb dues claus foranes referenciant a

*Fabricant* i *Cotxe* dins de la taula *Component*, ja que és la que té la multiplicitat superior en ambdós casos.

La relació *Telemetria* serà la més gran amb diferència, i cal adaptar els tipus de dades que farà servir als requeriments. És per això que el codi de les dades serà autoincremental i de mida bigint, i la precisió de l'atribut "moment" sera un datetime amb segons fraccionats (precisió 6a la definició de datetime) per desar fins als mil·lisegons. La relació *Component-Telemetria* té multiplicitat molts a 1 ja que un mateix component envia multiples dades i aquesta dada només pot pertànyer a un component. Aquesta relació és representa mitjançant una clau forana del component representat a cada fila de la taula *Telemetria*.

**País** (codi\_país, nom)

- codi\_país (PK): varchar(2) (codi ISO del país).
- nom: varchar(50).

**Persona** (codi\_persona, nom, nacionalitat)

- codi\_persona (PK): integer (mediumint), unsigned, NotNull, autoincrement
- nom: varchar(100).
- nacionalitat (FK): representa a *País*.

**Director** (codi\_persona, codi\_equip, any\_inici, any\_fi, estudis)

- codi\_persona (PK,FK): representa a *Persona*.
- codi equip (FK): representa a *Equip*
- any\_inici, any\_fi: year.
- estudis: text

**Pilot** (codi\_persona, equip , codi\_llicència, any\_inici, any\_fi,)

- codi\_persona (PK,FK): representa a *Persona*.
- equip (FK): representa a l'equip (codi equip).
- codi\_llicència: varchar(8)
- any\_inici, any\_fi: year.

**Patrocinador** (codi\_patrocinador, sector, nom)

- codi\_patrocinador (PK):integer (mediumint).
- sector (FK): Representa a *Sector* (codi\_sector).

**Sector** (codi\_sector, nom)

- codi\_sector (PK): integer (smallint).

**Equip** (codi equip, nom, seu, data\_debut)

- codi equip (PK): integer (smallint).
- nom: varchar(50).
- seu: varchar(100).

**Patrocini** (patrocinador, equip, cost,data\_inici, data\_fi)



- patrocinador, equip (PK).
- patrocinador (FK): representa a *Patrocinador* (codi\_patrocinador).
- equip (FK): representa a *Equip* (codi\_equip).
- cost: integer (int).
- any\_inici, any\_fi: year.

**Cotxe** (codi\_cotxe, model, any\_debut)

- codi\_cotxe (PK): integer (mediumint).
- model: varchar(25).
- any\_debut: year.

**Cotxe\_Equip** (codi\_cotxe, codi\_equip)

- codi\_cotxe, codi\_equip (PK).
- codi\_cotxe (FK): representa a *Cotxe* (codi\_cotxe)
- codi\_equip (FK): representa a *Equip* (codi\_equip).

**Component** (codi\_component, fabricant, cotxe, descripció, utilitat)

- codi\_component (PK): integer (mediumint).
- fabricant (FK): representa a *Fabricant* (codi\_fabricant).
- cotxe (FK): representa a *Cotxe* (codi\_cotxe).
- descripcio: text.
- utilitat: text.

**Fabricant** (codi\_fabricant, nom, data\_entrada)

- codi\_fabricant (PK): integer(mediumint).
- nom: varchar(50).
- any\_debut: year.

**Circuit** (codi\_circuit, nom, longitud, país)

- codi\_circuit (PK): integer (mediumint).
- nom: varchar(50)
- longitud: integer (smallint).
- pais (FK): representa a *País*.

**Competició** (codi\_competicio, temporada, nom, acronim, descripció)

- codi\_competicio (PK): integer (smallint).
- temporada: year.
- nom: varchar(50).
- acrònim: varchar(10).
- descripció: text.

**Emissora** (codi\_emissora, nom, país)

- codi\_emissora (PK): integer (smallint).
- nom: varchar(50).

- país: varchar(50).

#### **Retransmissió** (emissora, competicio)

- competicio, emissora (PK)
- competicio (FK): representa a *Competició* (codi\_categoria).
- emissora (FK): representa a *Emissora* (codi\_emissora).

#### **Cursa** (circuit, pilot, competició, posició, data\_cursa)

- circuit, pilot (PK).
- circuit (FK): representa a *Circuit* (codi\_circuit).
- pilot (FK): representa a *Pilot* (codi\_persona).
- competicio (FK): representa a *Competició* (codi\_competicio).
- posició: integer (smallint).
- data\_cursa: date.

#### **Telemetria** (codi\_dada, component, moment, àmbit, valor, unitat\_de\_mesura)

- codi\_dada (PK): integer (bigint).
- component (FK): representa a *Component* (codi\_component).
- moment: datetime(6).
- valor: float.
- unitat\_de\_mesura: varchar(25).

## **4.2 Normalització del disseny [17] [18]**

Per tal d'optimitzar el disseny creat per a la base de dades, en aquest capítol s'analitza si la transformació del model conceptual al relacional compleix amb les regles de normalització. Aquest procés pretén detectar possibles atributs redundants al disseny, problemes que puguin sorgir a l'hora d'actualitzar les taules i, en definitiva, millorar la integritat de les dades.

### **4.2.1 Primera forma normal**

Segons la definició<sup>4</sup>:

Una relació està en primera forma normal (1FN) si, i només si, cap atribut de la relació és ell mateix una relació, ni descomponible ni amb multiplicitat de valors. Els atributs, doncs, han de ser atòmics.

El disseny no presenta cap entitat amb atributs no atòmics, així que es pot considerar en aquest punt que compleix la primera forma normal.

### **4.2.2 Segona forma normal**

Segons la definició:

---

<sup>4</sup> Aquesta i les definicions de la 2NF, la 3NF i la FNBC han estat extretes del "Mòdul 3: Disseny lògic de bases de dades" dels apunts de l'assignatura Disseny de bases de dades.

Una relació està en segona forma normal (2FN) si, i només si, està en primera forma normal i tot atribut que no forma part d'una clau candidata depèn completament de totes les claus candidates de la relació.

Per tal de trobar entitats amb atributs que no compleixin la 2FN primer cal revisar aquelles entitats amb claus primàries no autoincrementals, com per exemple *Retransmissió* o *Cursa*, ja que per definició les altres estaran a la 2FN (la propia definició de clau primària fa que la resta d'atributs en depenguin i, si només té un sol atribut, la dependència serà completa). Un cop escollides les entitats candidates, es tracta de trobar atributs que depenguin d'una part de la clau primària.

Com a resultat de l'anàlisi, es pot concloure que cap entitat es troba fora de la 2FN, i es pot procedir a la següent forma normal.

### **4.2.3 Tercera forma normal**

Segons la definició:

Una relació està en tercera forma normal (3FN) si, i només si, està en segona forma normal i cap atribut que no forma part d'una clau candidata depèn d'un conjunt d'atributs que en conté algun que no forma part d'una clau candidata.

En aquest cas es troba de validar que cap entitat conté atributs redundants o que es poden calcular mitjançant altres atributs ja definits. Sembla que, també en aquest cas, es compleix la forma normal ja que no hi ha cap atribut que no s'hagi d'insertar manualment.

### **4.2.4 Forma normal de Boyce-Codd**

Segons la definició:

Una relació està en la forma normal de Boyce-Codd (FNBC) si, i només si, està en 3FN i els determinants de totes les dependències que presenta la relació en són claus candidates.

L'utilització de claus primàries autoincrementals en la majoria de taules fa difícil que es doni aquest cas. La resta, com *Cursa*, no presenta redundància de clau primària que faci possible la forma normal de Boyce-Codd

## 5. Disseny físic

### 5.1 Transformació del model lògic al model físic

Seguint el format descrit del disseny lògic, s'han creat les taules al sistema fent servir el codi de creació definit a l'arxiu `CREAR_TAULES.sql`. S'ha intentat ajustar el tipus de dades utilitzat per de guanyar eficiència i espai. Així, sempre que ha estat possible fer servir el tipus d'enter smallint o mediumint, s'ha optat per aquesta opció en comptes de l'enter per defecte. De la mateixa manera, si una variable de tipus cadena farà servir un màxim de caràcters per a la seva definició, s'ha especificat aquest límit al tipus de dades (per exemple, el codi de país només fa servir dues lletres, per tant, s'ha definit el tipus de dades `varchar(2)` en aquest cas concret).

### 5.2 Procediments d'ABM i disparadors.

La base de dades serà independent de la aplicació de gestió, per això s'han creat una sèrie de procediment emmagatzemats per tal de poder realitzar les funcions d'alta, baixa i modificació (ABM) a les taules del disseny.

En quant a les restriccions que cal implementar per tal de comprovar que les dades que s'introdueixen a la base de dades, es farà també majoritàriament mitjançant procediments emmagatzemats i disparadors. Altres SGBD fan servir la opció *check* a la definició de taules per tal de restringir les insercions o modificacions en funció de regles concretes, però MySQL no permet encara aquesta funcionalitat.

Per controlar més acuradament com s'esborren les dades del sistema, s'ha decidit deshabilitar per defecte la opció *on delete cascade*. Els procediments s'encarregaran de mantenir la coherència entre les claus foranes de les diferents taules. Així, si per exemple s'esborra un cotxe del sistema, el procediment esborrarà primer aquelles entrades que fan referència al cotxe i que en depenen directament.

Per últim comentar que s'han implementat tots els procediments seguint les descripcions requeriments del disseny. Així, tots comptaran amb una cadena com a paràmetre de sortida que mostrarà el resultat de l'execució del procediment.

## 5.2.1 Procediments d'alta.

Seguidament es descriuen cadascun dels procediments d'alta implementats mitjançant una descripció dels arguments fets servir i d'aquelles restriccions que es tenen en compte a l'hora de donar d'alta un nou element.

### 5.2.1.1 Alta circuit

Dona d'alta un circuit al sistema. A la [Fig. 5.1](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nom o sobrenom que ja existeix actualment a la taula *circuit*.
- Nacionalitat no definida a la taula *país*.
- Longitud menor a 1000 metres (disparador *chk\_circuit\_long\_insert*).

nom	VARCHAR(50)	➔ IN
sobrenom	VARCHAR(50)	➔ IN
longitud	SMALLINT	➔ IN
nacionalitat	CHAR(2)	➔ IN
RSP	VARCHAR(512)	➔ OUT

Fig. 5.1 Paràmetres del procediment *alta\_circuit*

### 5.2.1.2 Alta competició

Dona d'alta una competició al sistema. A la [Fig. 5.2](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nom o acrònim que ja existeix actualment a la taula *competició*.

nom	VARCHAR(50)	➔ IN
acronim	VARCHAR(10)	➔ IN
descripcio	TEXT	➔ IN
RSP	VARCHAR(250)	➔ OUT

Fig. 5.2 Paràmetres del procediment *alta\_competició*

### 5.2.1.3 Alta component

Dona d'alta un component al sistema. A la [Fig. 5.3](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Fabricant no definit a la taula *fabricant*.

- Cotxe no definit a la taula *cotxe*.

nom_fabricant	VARCHAR(50)	➔ IN
model_cotxe	VARCHAR(25)	➔ IN
descripcio	TEXT	➔ IN
utilitat	TEXT	➔ IN
RSP	VARCHAR(512)	➤ OUT

Fig. 5.3 Paràmetres del procediment *alta\_component*

#### 5.2.1.4 Alta cotxe

Dona d'alta un cotxe al sistema. A la [FIG. 5.4](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Model que ja existeix actualment a la taula *cotxe*.
- Codi de l'equip no definit a la taula *equip*

model	VARCHAR(25)	➔ IN
any_debut	YEAR	➔ IN
in equip	MEDIUMINT	➔ IN
temporada	YEAR	➔ IN
RSP	VARCHAR(250)	➤ OUT

Fig. 5.4 Paràmetres del procediment *alta\_cotxe*

#### 5.2.1.5 Alta cursa

Dona d'alta una cursa al sistema. A la [FIG. 5.5](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Combinació de circuit, pilot i competició ja existents a la taula *cursa*.
- Sobrenom del circuit no definit a la taula *circuit*.
- Pilot no definit a la taula *Pilot*.
- Competició no definida a la taula *Competició*.
- L'any de la data de la cursa no es correspon amb la temporada de la competició definida a la taula *competició\_temporada*.

sobrenom_circuit	VARCHAR(50)	➡ IN
nom_pilot	VARCHAR(100)	➡ IN
nom_competicio	VARCHAR(50)	➡ IN
posicio	SMALLINT	➡ IN
data_cursa	DATE	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.5 Paràmetres del procediment alta\_cursa

### 5.2.1.6 Alta director

Dona d'alta un director al sistema. Com *director* és una subclasse de *persona*, si la persona no està definida al sistema caldrà crear-la abans. A la [FIG. 5.6](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Director ja assignat actualment a un altre equip.
- País no definit a la taula pais.
- Nom d'equip no definit a la taula equip

in_nom	VARCHAR(100)	➡ IN
in_pais	CHAR(2)	➡ IN
in_equip	MEDIUMINT	➡ IN
in_estudis	TEXT	➡ IN
in_debut	YEAR	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.6 Paràmetres del procediment alta\_director

### 5.2.1.7 Alta emissora

Dona d'alta una emissora al sistema. A la [FIG. 5.7](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Parella emissora-país ja existent a la taula *emissora*.
- Nacionalitat no definida a la taula *país*.

nom	VARCHAR(50)	➡ IN
nacionalitat	CHAR(2)	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.7 Paràmetres del procediment alta\_emissora

### 5.2.1.8 Alta equip

Dona d'alta un equip al sistema. A la [FIG. 5.9](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nom de l'equip ja definit a la taula *equip*.

nom	VARCHAR(50)	➡ IN
seu	VARCHAR(100)	➡ IN
debut	DATE	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.8 Paràmetres del procediment *alta equip*

### 5.2.1.9 Alta fabricant

Dona d'alta un fabricant al sistema. A la [Fig. 5.10](#)[FIG. 5.9](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nom del fabricant ja definit a la taula *fabricant*.

nom	VARCHAR(50)	➡ IN
debut	YEAR	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.9 Paràmetres del procediment *alta fabricant*

### 5.2.1.10 Alta país

Dona d'alta un país al sistema. A la [FIG. 5.10](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nom del país ja definit a la taula *país*.

codi	CHAR(2)	➡ IN
nom	VARCHAR(50)	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.10 Paràmetres del procediment *alta país*

### 5.2.1.11 Alta patrocinador

Dona d'alta un patrocinador al sistema. A la [FIG. 5.11](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:



- Nom del patrocinador ja definit a la taula *patrocinador*.
- Sector no definit a la taula *sector*.

sector_patrocinador	SMALLINT	➔ IN
nom_patrocinador	VARCHAR(50)	➔ IN
RSP	VARCHAR(250)	➡ OUT

Fig. 5.11 Paràmetres del procediment *alta\_patrocinador*

#### 5.2.1.12 Alta patrocini

Dona d'alta una relació de patrocini al sistema. S'ha limitat a 10.000 el cost mínim d'un patrocini per tal d'evitar possibles entrades errònies degut a problemes d'unitats (si l'usuari es pensa que el cost es troba en milions, per exemple, el valor introduït serà refusat ja que probablement introduirà un valor menor a 10.000). A la [FIG. 5.12](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Combinació de patrocinador, equip i temporada ja existent a la taula *patrocini*.
- Nom del patrocinador no definit a la taula *patrocinador*.
- Nom de l'equip no definit a la taula *equip*.
- Cost del patrocini menor a 10.000 (disparador *chk\_circuit\_long\_insert*).

in_patrocinador	MEDIUMINT	➔ IN
in_equip	SMALLINT	➔ IN
temporada	YEAR	➔ IN
cost	INT	➔ IN
RSP	VARCHAR(250)	➡ OUT

Fig. 5.12 Paràmetres del procediment *alta\_patrocini*

#### 5.2.1.13 Alta pilot

Dona d'alta un pilot al sistema. Com *pilot* és una subclasse de *persona*, si la persona no està definida al sistema caldrà crear-la abans A la [FIG. 5.13](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nom del pilot ja definit a la taula *pilot*.
- Nom de l'equip no definit a la taula *equip*.
- Nacionalitat no definida a la taula *país*.

nom	VARCHAR(100)	➡ IN
nacionalitat	CHAR(2)	➡ IN
in_equip	MEDIUMINT	➡ IN
llicencia	VARCHAR(8)	➡ IN
inici	YEAR	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.13 Paràmetres del procediment alta\_pilot

#### 5.2.1.14 Alta retransmissió

Dona d'alta una retransmissió al sistema. A la [FIG. 5.14](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nacionalitat no definida a la taula *país*.
- Competició no definida a la taula *competició*.
- Emissora no definida a la taula *emissora*.
- Si la parella emissora-país no està definida a la taula *emissora* (això significa que la emissora no emet al país passat com a paràmetre)..

in_emissora	MEDIUMINT	➡ IN
pais_emissio	CHAR(2)	➡ IN
in_competicio	VARCHAR(50)	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.14 Paràmetres del procediment alta\_retransmissió

#### 5.2.1.15 Alta sector

Dona d'alta un sector al sistema. A la [FIG. 5.15](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Nom del sector ja definit a la taula sector.

nom	VARCHAR(50)	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.15 Paràmetres del procediment alta\_sector

#### 5.2.1.16 Alta telemetria

Dona d'alta una dada telemètrica al sistema. A la [FIG. 5.16](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

-

- Component no definit a la taula component.
- Velocitat més gran de 650 km/h. (Trigger *chk\_telemetria\_value\_insert*)

in_component	MEDIUMINT	➔ IN
in_moment	DATETIME(6)	➔ IN
in_ambit	VARCHAR(50)	➔ IN
in_valor	FLOAT	➔ IN
in_unitats	VARCHAR(25)	➔ IN
RSP	VARCHAR(512)	➡ OUT

Fig. 5.16 Paràmetres del procediment *alta\_telemetria*

### 5.2.1.17 Alta temporada

Dona d'alta un país al sistema. A la [FIG. 5.17](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Competició no definida a la taula *competició*.
- Parella competició – temporada ja definida a la taula *competició\_temporada*.

any_nou	YEAR	➔ IN
in_competicio	SMALLINT	➔ IN
RSP	VARCHAR(250)	➡ OUT

Fig. 5.17 Paràmetres del procediment *alta\_temporada*

## 5.2.2 Procediments de baixa.

### 5.2.2.1 Baixa circuit

Dona de baixa un circuit al sistema així com aquelles curses amb el circuit com a clau forana (taula *cursa*). A la [FIG. 5.18](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del circuit no definit a la taula circuit.

codi	MEDIUMINT	➔ IN
RSP	VARCHAR(250)	➡ OUT

Fig. 5.18 Paràmetres del procediment *baixa\_circuit*

### 5.2.2.2 Baixa competició

Dona de baixa una competició al sistema així com les files referenciades a les taules *competició\_temporada*, *retransmissió* i *cursa*. A la [FIG. 5.19](#) es mostren els paràmetres del

mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la competició no definit a la taula competició.

any_nou	YEAR	➡ IN
nom_competicio	VARCHAR(50)	➡ IN
RSP	VARCHAR(250)	⬅ OUT

Fig. 5.19 Paràmetres del procediment baixa\_competició

### 5.2.2.3 Baixa component

Dona de baixa un component al sistema així com les files referenciades a la taula *telemetry*. A la [Fig. 5.20](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del component no definit a la taula component.

codi	MEDIUMINT	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.20 Paràmetres del procediment baixa\_component

### 5.2.2.4 Baixa cotxe

Dona de baixa un cotxe al sistema així com les files referenciades a les taules *component* i *cotxe equip*. A la [Fig. 5.21](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del cotxe no definit a la taula cotxe.

codi	MEDIUMINT	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.21 Paràmetres del procediment baixa\_cotxe

### 5.2.2.5 Baixa cursa

Dona de baixa una cursa al sistema. A la [Fig. 5.22](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del circuit no definit a la taula *circuit*.
- Codi del pilot no definit a la taula *Pilot*.
- Codi de la competició no definida a la taula *Competició*.

in_circuit	MEDIUMINT	⇒ IN
in_pilot	MEDIUMINT	⇒ IN
in_competicio	SMALLINT	⇒ IN
RSP	VARCHAR(512)	⇐ OUT

Fig. 5.22 Paràmetres del procediment baixa\_cursa

### 5.2.2.6 Baixa director

Dona de baixa un director al sistema així com la referencia a la seva superclasse *persona*. A la [FIG. 5.23](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la persona no definit a la taula director.

in_persona	MEDIUMINT	⇒ IN
RSP	VARCHAR(512)	⇐ OUT

Fig. 5.23 Paràmetres del procediment baixa\_director

### 5.2.2.7 Baixa emissora

Dona de baixa una emissora al sistema així com les retransmissions referenciades si existissin (taula *retransmissió*). A la [FIG. 5.24](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de l'emissora no definit a la taula emissora.

in_emissora	SMALLINT	⇒ IN
RSP	VARCHAR(512)	⇐ OUT

Fig. 5.24 Paràmetres del procediment baixa\_emissora

### 5.2.2.8 Baixa equip

Dona de baixa un equip al sistema així com les files referenciades a les taules *director*, *pilot*, *cotxe* i *patrocini*. A la [FIG. 5.25](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de l'equip no definit a la taula equip.

in equip	SMALLINT	⇒ IN
RSP	VARCHAR(512)	⇐ OUT

Fig. 5.25 Paràmetres del procediment baixa equip

### 5.2.2.9 Baixa fabricant

Dona de baixa un fabricant al sistema així com els components relatius a aquest fabricant (i, per tant, la telemetria relativa a aquests components). A la [FIG. 5.26](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del fabricant no definit a la taula fabricant.

in_fabricant	MEDIUMINT	➔ IN
RSP	VARCHAR(512)	➜ OUT

Fig. 5.26 Paràmetres del procediment baixa\_fabricant

### 5.2.2.10 Baixa país

Dona de baixa un país al sistema però no així les files referenciades a altres taules, ja que l'atribut país no forma part d'una relació de dependència amb les altres entitats. A la [FIG. 5.27](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del país no definit a la taula país.

in_país	CHAR(2)	➔ IN
RSP	VARCHAR(512)	➜ OUT

Fig. 5.27 Paràmetres del procediment baixa\_país

### 5.2.2.11 Baixa patrocinador

Dona de baixa un patrocinador al sistema així com els patrocinis relatius a la taula *patrocini*. A la [FIG. 5.28](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del patrocinador no definit a la taula patrocinador.

in_patrocinador	MEDIUMINT	➔ IN
RSP	VARCHAR(512)	➜ OUT

Fig. 5.28 Paràmetres del procediment baixa\_patrocinador

### 5.2.2.12 Baixa patrocini

Dona de baixa una relació de patrocini al sistema. A la [FIG. 5.29](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del patrocinador no definit a la taula *patrocinador*.
- Codi de l'equip no definit a la taula *equip*.
- Temporada no definida a la taula *temporada*.

in_patrocinador	MEDIUMINT	➔ IN
in equip	SMALLINT	➔ IN
in_temporada	YEAR	➔ IN
RSP	VARCHAR(512)	➔ OUT

Fig. 5.29 Paràmetres del procediment baixa\_patrocini

### 5.2.2.13 Baixa pilot

Dona de baixa un pilot del sistema així com la seva entrada a la seva superclasse (taula *persona*). A la [FIG. 5.30](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la persona no definit a la taula *pilot*.

in_persona	MEDIUMINT	➔ IN
RSP	VARCHAR(512)	➔ OUT

Fig. 5.30 Paràmetres del procediment baixa\_pilot

### 5.2.2.14 Baixa retransmissió

Dona de baixa una retransmissió al sistema. A la [FIG. 5.31](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la competició no definit a la taula *competició*.
- Codi de l'emissora no definit a la taula *emissora*.

in_emissora	SMALLINT	➔ IN
in_competicio	SMALLINT	➔ IN
RSP	VARCHAR(512)	➔ OUT

Fig. 5.31 Paràmetres del procediment baixa\_retransmissió

### 5.2.2.15 Baixa sector

Dona de baixa un sector al sistema. A la [FIG. 5.32](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del sector no definit a la taula *sector*.

in_sector	SMALLINT	➔ IN
RSP	VARCHAR(512)	➡ OUT

Fig. 5.32 Paràmetres del procediment *baixa\_sector*

### 5.2.2.16 Baixa telemetria

Dona de baixa una dada telemètrica del sistema. A la [FIG. 5.33](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la dada no definit a la taula *telemetria*.

in_dada	BIGINT	➔ IN
RSP	VARCHAR(512)	➡ OUT

Fig. 5.33 Paràmetres del procediment *baixa telemetria*

### 5.2.2.17 Baixa temporada

Dona de baixa una temporada al sistema, però no esborra les claus foranes referenciades sinó que llençarà un error. A la [FIG. 5.34](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Any no definit a la taula *temporada*.
- Codi de la competició no definit a la taula *competició*.

in_temporada	YEAR	➔ IN
in_competicio	SMALLINT	➔ IN
RSP	VARCHAR(512)	➡ OUT

Fig. 5.34 Paràmetres del procediment *baixa temporada*

## 5.2.3 Procediments de modificació

### 5.2.3.1 Modificació circuit

Modifica un circuit del sistema. A la [FIG. 5.35](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del circuit no definit a la taula *circuit*.
- Els errors descrits a l'apartat [ALTA CIRCUIT](#) (veure 5.2.1.1).



in_codi	MEDIUMINT	➔ IN
in_nom	VARCHAR(50)	➔ IN
in_sobrenom	VARCHAR(50)	➔ IN
in_longitud	SMALLINT	➔ IN
in_nacionalitat	CHAR(2)	➔ IN
RSP	VARCHAR(512)	➡ OUT

Fig. 5.35 Paràmetres del procediment modificació\_circuit

### 5.2.3.2 Modificació competició

Modifica una competició del sistema. A la [Fig. 5.36](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la competició no definit a la taula *competició*.
- Els errors descrits a l'apartat [FIG. 5.1](#) Paràmetres del procediment alta\_circuit
- **Alta competició** (veure 5.2.1.2).

in_codi	SMALLINT	➔ IN
in_nom	VARCHAR(50)	➔ IN
in_acronim	VARCHAR(10)	➔ IN
in_descripcio	TEXT	➔ IN
RSP	VARCHAR(512)	➡ OUT

Fig. 5.36 Paràmetres del procediment modificació\_competició

### 5.2.3.3 Modificació component

Modifica un component del sistema. A la [Fig. 5.37](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del component no definit a la taula *component*.
- Els errors descrits a l'apartat [Fig. 5.2](#) Paràmetres del procediment alta\_competició
- **Alta component** (veure 5.2.1.3).

in_codi	SMALLINT	➔ IN
in_fabricant	MEDIUMINT	➔ IN
in_cotxe	MEDIUMINT	➔ IN
in_descripcio	TEXT	➔ IN
in_utilitat	TEXT	➔ IN
RSP	VARCHAR(512)	➡ OUT

Fig. 5.37 Paràmetres del procediment modificació\_component

### 5.2.3.4 Modificació cotxe

Modifica un cotxe del sistema. A la [FIG. 5.38](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi del cotxe no definit a la taula *cotxe*.
- Els errors descrits a l'apartat [FIG. 5.3](#) Paràmetres del procediment *alta\_cotxe*
- **Alta cotxe** (veure 5.2.1.4).

in_codi	MEDIUMINT	➔ IN
in_model	VARCHAR(25)	➔ IN
in_any	YEAR	➔ IN
RSP	VARCHAR(512)	➔ OUT

Fig. 5.38 Paràmetres del procediment *modificació\_cotxe*

### 5.2.3.5 Modificació cursa

Modifica una cursa del sistema. A la [FIG. 5.39](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi dels elements de la clau primària (circuit, pilot, competició, data\_cursa) no definit a la taula respectiva.
- Els errors descrits a l'apartat [FIG. 5.4](#) Paràmetres del procediment *alta\_cotxe*
- **Alta cursa** (veure 5.2.1.5).

in_circuit	MEDIUMINT	➔ IN
in_pilot	MEDIUMINT	➔ IN
in_competicio	SMALLINT	➔ IN
in_posicio	SMALLINT	➔ IN
in_data_cursa	DATE	➔ IN
RSP	VARCHAR(512)	➔ OUT

Fig. 5.39 Paràmetres del procediment *modificació\_cursa*

### 5.2.3.6 Modificació director

Modifica un director del sistema així com els atributs de la classe superior *Persona*. A la [FIG. 5.40](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la persona no definit a la taula *director*.
- Els errors descrits a l'apartat [FIG. 5.5](#) Paràmetres del procediment *alta\_cursa*
- **Alta director** (veure 5.2.1.6).

in_codi	MEDIUMINT	⇒ IN
in_nom	VARCHAR(100)	⇒ IN
in_pais	CHAR(2)	⇒ IN
in equip	SMALLINT	⇒ IN
in_estudis	TEXT	⇒ IN
in_inici	YEAR	⇒ IN

Fig. 5.40 Paràmetres del procediment modificació\_director

### 5.2.3.7 Modificació emissora

Modifica una emissora del sistema. A la [FIG. 5.41](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi no definit a la taula *emissora*.
- Els errors descrits a l'apartat [Fig. 5.6](#) Paràmetres del procediment alta\_director
- **Alta emissora** (veure 5.2.1.7).

in_codi	SMALLINT	⇒ IN
in_nom	VARCHAR(50)	⇒ IN
in_pais	CHAR(2)	⇒ IN
RSP	VARCHAR(512)	⇐ OUT

Fig. 5.41 Paràmetres del procediment modificació\_emissora

### 5.2.3.8 Modificació equip

Modifica un equip del sistema. A la [FIG. 5.42](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi no definit a la taula *equip*.
- Els errors descrits a l'apartat [FIG. 5.7](#) Paràmetres del procediment alta\_emissora

- **Alta equip** (veure 5.2.1.8).

in_codi	SMALLINT	➡ IN
in_nom	VARCHAR(50)	➡ IN
in_seu	VARCHAR(100)	➡ IN
in_debut	DATE	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.42 Paràmetres del procediment modificació\_equip

### 5.2.3.9 Modificació fabricant

Modifica un fabricant del sistema. A la [FIG. 5.43](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi no definit a la taula fabricant.
- Els errors descrits a l'apartat [FIG. 5.8](#) Paràmetres del procediment alta equip
- **Alta fabricant** (veure 5.2.1.9).

in_codi	MEDIUMINT	➡ IN
in_nom	VARCHAR(50)	➡ IN
in_debut	YEAR	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.43 Paràmetres del procediment modificació\_fabricant

### 5.2.3.10 Modificació país

Modifica un país del sistema. A la [FIG. 5.44](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi no definit a la taula país.
- Els errors descrits a l'apartat [FIG. 5.9](#) Paràmetres del procediment alta\_fabricant
- **Alta país** (veure 5.2.1.10).

in_codi	CHAR(2)	➡ IN
in_nom	VARCHAR(50)	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.44 Paràmetres del procediment modificació\_país

### 5.2.3.11 Modificació patrocinador

Modifica un patrocinador del sistema. A la [FIG. 5.45](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi no definit a la taula patrocinador.
- Els errors descrits a l'apartat [FIG. 5.10](#) Paràmetres del procediment alta\_país
- **Alta patrocinador** (veure 5.2.1.11).

in_codi	MEDIUMINT	➡ IN
in_sector	SMALLINT	➡ IN
in_nom	VARCHAR(50)	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.45 Paràmetres del procediment modificació\_patrocinador

### 5.2.3.12 Modificació patrocini

Modifica un patrocini del sistema. A la [FIG. 5.46](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi dels elements de la clau primària (patrocinador, equip i temporada) no definits a la taula respectiva.
- Els errors descrits a l'apartat [FIG. 5.11](#) Paràmetres del procediment alta\_patrocinador
- **Alta patrocini** (veure 5.2.1.12).

in_patrocinador	MEDIUMINT	➡ IN
in_equip	SMALLINT	➡ IN
in_temporada	YEAR	➡ IN
in_cost	INT	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.46 Paràmetres del procediment modificació\_patrocini

### 5.2.3.13 Modificació pilot

Modifica un pilot del sistema així com els atributs de la classe superior *Persona*. A la [FIG. 5.47](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi de la persona no definit a la taula *pilot*.
- Els errors descrits a l'apartat [FIG. 5.12](#) Paràmetres del procediment alta\_patrocini
- **Alta pilot** (veure 5.2.1.13).

in_persona	MEDIUMINT	➡ IN
in_nom	VARCHAR(100)	➡ IN
in_pais	CHAR(2)	➡ IN
in_equip	SMALLINT	➡ IN
in_licencia	VARCHAR(8)	➡ IN
in_inici	YEAR	➡ IN

Fig. 5.47 Paràmetres del procediment modificació\_pilot

### 5.2.3.14 Modificació sector

Modifica un sector del sistema. A la [FIG. 5.48](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi no definit a la taula *sector*.
- Els errors descrits a l'apartat [FIG. 5.14](#) Paràmetres del procediment *alta\_retransmissió*
- **Alta sector** (veure 5.2.1.14).

in_sector	SMALLINT	➡ IN
in_nom	VARCHAR(50)	➡ IN
RSP	VARCHAR(512)	⬅ OUT

Fig. 5.48 Paràmetres del procediment *modificació\_sector*

### 5.2.3.15 Modificació telemetria

Modifica un dada telemètrica del sistema. A la [FIG. 5.49](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Codi no definit a la taula *telemetria*.
- Els errors descrits a l'apartat [FIG. 5.15](#) Paràmetres del procediment *alta\_sector*
- **Alta telemetria** (veure 5.2.1.15).

in_dada	BIGINT	➡ IN
in_component	MEDIUMINT	➡ IN
in_moment	DATETIME(6)	➡ IN
in_ambit	VARCHAR(50)	➡ IN
in_valor	FLOAT	➡ IN
in_unitats	VARCHAR(25)	➡ IN

Fig. 5.49 Paràmetres del procediment *modificació\_telemetria*

### 5.2.3.16 Modificació temporada

Modifica una temporada del sistema. A la [FIG. 5.50](#) es mostren els paràmetres del mètode amb el tipus de dades corresponents. Les següents restriccions produiran errors en la execució del procediment si no es compleixen:

- Parella any i competició no definida a la taula *competició\_temporada*.
- Els errors descrits a l'apartat [FIG. 5.16](#) Paràmetres del procediment *alta\_telemetria*
- **Alta temporada** (veure 5.2.1.16).

in_old_any	YEAR	➔ IN
in_old_competicio	SMALLINT	➔ IN
in_nou_any	YEAR	➔ IN
in_nou_competicio	SMALLINT	➔ IN
RSP	VARCHAR(250)	➔ OUT

Fig. 5.50 Paràmetres del procediment modificació\_temporada

## 5.2.4 Disparadors

Per evitar errors a l'hora d'introduir dades al sistema s'han creat una serie de disparadors que eviten la inserció o la actualització en certs casos si les dades no compleixen les condicions. En cas d'error serà el procediment d'alta o modificació el que s'encarregarà de mostrar la descripció mitjançant el paràmetre de sortida RSP.

- Longitud del circuit: es considera que un circuit no pot tenir una longitud menor a 1000 metres. Disparadors chk\_circuit\_long\_insert i chk\_circuit\_long\_update.
- Cost del patrocini: es considera que un patrocini sempre serà major a 10.000 euros. Disparadors chk\_cost\_patrocini\_insert i chk\_cost\_patrocini\_update.
- Velocitat reportada per un component telemètric massa gran: Es considera que un cotxe no pot anar a més de 500 km/h. Disparadors chk\_telemetria\_value\_insert i chk\_telemetria\_value\_update.

## 5.3 Log i control d'errors.

El sistema encapsula les funcions d'ABM mitjançant els procediments emmagatzemats descrits anteriorment. Per tal de realitzar un control d'errors adient amb l'encapsulació, també són els procediments els que controlen les senyals d'errors que es produeixen.

També s'ha implementat com a part del sistema de control d'errors un log que registra totes aquelles operacions d'ABM que es produeixen el sistema fent servir els procediments emmagatzemats.

### 5.3.1 Sistema de log.

El sistema de log es basa en una taula anomenada LOG (veure figura [FIG. 5.51](#)) en la qual s'escriuen totes les operacions efectuades pels procediments d'ABM. Concretament es desarà la següent informació després de cada operació:

- Tipus d'operació: Aquest camp descriu quin és el tipus d'operació efectuada i només podrà prendre els valors 'ALTA', 'BAIXA' o 'MODIFICACIÓ'.
- Taula afectada: Mostra el nom de la taula involucrada a l'operació.
- Valor antic: Aquest camp mostra quins eren els valor de la fila previs a l'operació d'ABM.
- Valor nou: D'igual manera, es mostren els valors que té la fila un cop finalitzada la operació d'ABM.



- Data: En aquest camp es desa la data i l'hora en que l'operació ha tingut lloc.
- Consulta: Per finalitzar es mostra la consulta completa que ha produït la modificació.



Fig. 5.51 Representació UML de la taula LOG

Els encarregats d'escriure a aquesta taula són els propis procediments emmagatzemats que, un cop comprovat que no s'ha produït cap error i que la operació s'ha dut a terme, realitzaran la inserció corresponent.

### 5.3.2 Control d'errors.

El control d'errors implementat es fa càrrec dels missatges d'error que d'una altra manera s'haurien mostrat per la sortida estàndard i els passa com a paràmetre de sortida. Són els propis procediments emmagatzemats els encarregats durant la seva execució d'agafar les senyals d'error i aplicar el format adient de sortida.

A més d'aquesta funció, també es fa servir aquest control d'errors per tal de comprovar que les dades introduïdes són adequades a part de ser del tipus correcte. Aquesta funció és la que a d'altres SGBD s'anomena CHECK i es defineix a la definició de les taules. En MySQL però, aquesta funció no està disponible, així que s'ha implementat mitjançant l'ús dels *handlers* a dins dels procediments d'ABM.

Els *handlers* definits són els següents:

- Senyal **1062**: Correspon a un error produït per na inconsistència en la clau primària. Es produeix normalment perquè s'intenta introduir un valor amb una clau primària preexistent a la taula destí.
- Senyal **1644**: Fa referència a un error comunicat per un disparador el qual no ha considerat com a vàlid algun valor que s'ha intentat introduir.
- Senyal **1452**: Correspon a un error produït per una inconsistència de clau forana. Es produeix normalment perquè s'ha intentat introduir un valor a un camp que és clau forana i aquest no existeix a la taula referenciada.
- Senyal **SQLEXCEPTION**: Engloba tots aquells errors que no s'han considerat a cap *handler*. D'aquesta manera no es llençaran per la sortida estàndard tot i no haver-los considerat.

- Senyal **SQLWARNING**: Fa la mateixa funció que la SQLEXCEPTION però amb els missatges d'avís en comptes dels d'error.

## 6. Mòdul estadístic

Una de les parts que aporten més valor útil a aquest projecte és el repositori estadístic. Mitjançant la creació d'un *data warehouse* es dona resposta al requeriment de generar estadístiques concretes que es puguin accedir amb un temps constant 1. Amb tal propòsit, s'ha dissenyat un model basat en una única taula de fets i una sèrie de taules complementàries que s'actualitzen mitjançant els procediments ABM. D'aquesta manera, el mòdul es mantindrà actualitzat en tot moment.

### 6.1 Disseny conceptual

La taula principal del mòdul estadístic (la també anomenada taula de fets) desarà només les dades finals que seran proporcionades pel mòdul. Les taules complementàries són necessàries per tal de mantenir l'ordre de cadascuna de les classificacions i així poder actualitzar la taula de fets amb aquells camps que ocupen certes posicions dins d'aquestes taules.

La següent classificació de les estadístiques requerides servirà com a taula de referència del mòdul estadístic per a la resta d'apartats:

- **Mòdul 1:** Percentatge de components defectuosos per cada cotxe.
- **Mòdul 2:** Resultat de cada pilot per competició. Es considera competició a cada categoria disputada per temporada.
- **Mòdul 3:** Els 10 pilots amb més curses finalitzades.
- **Mòdul 4:** Classificació per competició. Novament es considera competició a cada categoria disputada per temporada.
- **Mòdul 5:** Les 10 voltes més ràpides donades per circuit.
- **Mòdul 6:** Temperatura més alta enregistrada.
- **Mòdul 7:** Els 5 patrocinadors que més diners aporten.
- **Mòdul 8:** Volum de dades telemètriques i equip que més dades ha enregistrat.
- **Mòdul 9:** Els 5 fabricants que més components aporten per competició.
- **Mòdul 10:** Els 10 pilots que més curses han guanyat.
- **Mòdul 11:** Pilot amb la millor evolució.
- **Mòdul 12:** Cotxe que més carburant ha consumit per competició i temporada.

#### 6.1.1 Taula de fets

La taula referència les següents claus foranes per tal de donar resposta als mòduls estadístics demanats:

- Competició: FK de la taula *competició* (codi\_competicio).
- Temporada: FK de la taula *temporada* (any).

- Pilot: FK de la taula *pilot* (codi\_persona).
- Equip: FK de la taula *equip* (codi\_equip).
- Cotxe: FK de la taula *cotxe* (codi\_cotxe). Necessari
- Circuit: FK de la taula *circuit* (codi\_circuit).
- Data cursa: FK de la taula *cursa* (data\_cursa).
- Patrocinador: FK de la taula *patrocinador* (codi\_patrocinador).
- Fabricant: FK de la taula *fabricant* (codi\_fabricant).

Es desaran els següents atributs, necessaris per a proporcionar les estadístiques requerides:

- Quantitat components: nombre de components fent referència al mòdul 9.
- Cost patrocini: quantitat aportada pel patrocinador fent referència al mòdul 7.
- Volta ràpida: temps per volta fent referència al mòdul 5.
- Temperatura màxima: valor de temperatura fent referència al mòdul 6.
- Dades telemètriques: nombre de dades fent referència al mòdul 8.
- Consum cotxe: valor del consum fent referència al mòdul 12.
- Punts pilot: nombre de punts fent referència al mòdul 4.
- Posició pilot: posició fent referència al mòdul 2.
- Curses finalitzades: nombre de curses fent referència al mòdul 3.
- Curses guanyades: nombre de curses fent referència al mòdul 10.
- Components defectuosos: percentatge de components fent referència al mòdul 1.

Per últim es desarà un atribut anomenat “camp de control” per tal de distingir quines files fan referència a quin mòdul. Aquest camp prendrà els següents valors depenent del mòdul al que es refereix:

- Mòdul 1: 0xCD (Components Defectuosos).
- Mòdul 2 i 4: 0xCP (Classificació Pilots).
- Mòdul 3: 0xCF (Curses Finalitzades).
- Mòdul 5: 0xVR (Volta Ràpida).
- Mòdul 6: 0xTM (Temperatura Màxima).
- Mòdul 7: 0xPA (Patrocinador).
- Mòdul 8: 0xTE (dades Telemètriques).
- Mòdul 9: 0xQC (Quantitat de Components).
- Mòdul 10: 0xCG (Curses Guanyades).
- Mòdul 11: 0xEP (Evolució de Pilot).
- Mòdul 12: 0xCC (Consum de Carburant).

## fact\_table\_1

```
competició: SMALLINT
temporada: YEAR
pilot: MEDIUMINT
equip: SMALLINT
cotxe: MEDIUMINT
circuit: MEDIUMINT
patrocinador: MEDIUMINT
fabricant: MEDIUMINT
data_cursa: DATE
data_millora: DATE
quantitat_components: MEDIUMINT
cost_patrocini: INT
volta_ràpida: TIME
temperatura_màxima: FLOAT(8,2)
dades_telemètriques: MEDIUMINT
dades_telemètriques_equip: MEDIUMINT
consum_cotxe: FLOAT(8,2)
punts_pilot: SMALLINT
posició_pilot: SMALLINT
curses_finalitzades: SMALLINT
curses_guanyades: SMALLINT
components_defectuosos: DECIMAL(4,1)
camp_control: CHAR(4)
```

Fig. 6.1 Representació UML de la taula de fets

### 6.1.2 Taules complementàries

A la taula de fets només es desaran els resultats estadístics finals. És a dir, si s'han de mostrar els 10 pilots amb més curses guanyades, a la taula de fets només es desaran aquestes files concretes. Es fan servir taules complementàries per a desar la classificació completa per a cada estadística així, seguint amb el mateix exemple, hi ha una taula complementària on es desen les curses guanyades per cada pilot ordenades per número de curses. D'aquesta manera que en tot moment es poden llegir les 10 primeres obtenint l'estadística desitjada.

Les taules complementàries necessàries per a complir els requeriments estadístics són les següents:

- **dim\_defectuós:** desa el codi del cotxe, el codi del component que pertany a aquest cotxe i si aquest es defectuós o no. Fa referència al mòdul estadístic 1.
- **dim\_classificació:** desa, per temporada i competició, el codi dels pilots i els punts acumulats d'aquest ordenats per punts. D'aquesta manera s'obté el resultat del mòdul estadístic 2 (ja que els pilots estan ordenats per posició) i

#### dim\_defectuós

```
cotxe: MEDIUMINT
component: MEDIUMINT
defectuós: BIT
```

#### dim\_classificacio

```
competició: SMALLINT
temporada: YEAR
pilot: MEDIUMINT
punts: SMALLINT
```

del mòdul 4.

- **dim\_resultat:** desa el codi del pilot juntament amb el nombre de curses guanyades i finalitzades. Agafant les 10 primeres files ordenades per nombre de curses de cada tipus, dona resposta als mòduls estadístics 3 i 10 respectivament..
- **dim\_volta\_ràpida:** desa els camps necessaris per resoldre el mòdul estadístic 5. Això és: codi del circuit, codi de l'equip, codi persona del pilot i data de la cursa (tots quatre com a clau primària d'una cursa), així com el temps per volta registrat.
- **dim\_patrocini:** desa els diferents patrocinadors amb la suma total dels seus patrocinis. Els 5 primers d'aquesta llista ordenada per el cost del patrocini total dona resposta al mòdul estadístic 7.
- **dim\_telemetria:** aquesta taula desa, per temporada i equip, les quantitat de dades telemètriques enregistrades. Aquest disseny permet d'una banda saber quin és l'equip en tot moment que més dades telemètriques ha enregistrat a un any concret i, fent el sumatori d'un any, les dades enregistrades aquest any. Ambdós valors són requeriments del mòdul estadístic 8.
- **dim\_component:** desa el codi del fabricant i els components relacionats enregistrats al sistema. D'aquesta manera, ordenant els resultat per quantitat de components i agafant els 5 primers, es dona resposta al mòdul estadístic 9.
- **dim\_consum:** aquesta taula desa, per competició i temporada, el consum acumulat de cada cotxe. Ordenant els resultats per competició, temporada i consum, i llegint el primer resultat de cada competició i temporada, és dona resposta al mòdul 12.

#### dim\_resultat

pilot: MEDIUMINT  
guanyades: SMALLINT  
finalitzades: SMALLINT

#### dim\_volta\_rapida

circuit: MEDIUMINT  
equip: SMALLINT  
pilot: MEDIUMINT  
data: DATE  
temps: TIME

#### dim\_patrocini

patrocinador: MEDIUMINT  
cost: INT

#### dim\_telemetria

temporada: YEAR  
equip: SMALLINT  
dades\_any: MEDIUMINT

#### dim\_component

fabricant: MEDIUMINT  
component: MEDIUMINT

#### dim\_consum

competició: SMALLINT  
temporada: YEAR  
cotxe: MEDIUMINT  
consum\_cotxe: FLOAT(8,4)

## 6.2 Disseny físic

Per tal de complir el requisit de mantenir la taula de fets actualitzada en tot moment, s'han implementat noves funcionalitat als procediments d'ABM de tal manera que, cada cop que es produeix un canvi al sistema, s'actualitzen les taules estadístiques immediatament.

Com a funcionalitat extra a part dels requeriments del projecte, es planteja el problema que en algun moment sigui necessari crear les estadístiques des d'un estat inicial i no sigui possible fer servir els procediments d'ABM. Per a solucionar aquesta situació, s'han creat una sèrie de procediments que recorren totes les taules del sistema i actualitzen les estadístiques des de zero. D'aquesta manera, si es perden les estadístiques o es dona la situació en que les taules tenen dades i volem incloure-les a les estadístiques, es poden executar aquests procediments en qualsevol moment i actualitzar els magatzem de dades. Un cop en aquest estat, els procediments d'ABM actualitzaran les estadístiques en endavant.

### 6.2.1 Actualització incremental

Cada mòdul estadístic serà actualitzat després de cada alta, baixa o modificació del sistema mitjançant els procediments emmagatzemats definits. L'objectiu serà que en qualsevol moment, la taula de fets i les taules complementàries a aquesta estiguin actualitzades. A continuació es descriuran quins són els procediments i les taules estadístiques afectades per tal de proporcionar una solució adient i permanentment actualitzada.

#### 6.2.1.1 Mòdul 1 (Percentatge de components defectuosos per cada cotxe)

El procediment *alta\_component* per defecte inserta el component a la taula *dim\_defectuos* considerant-lo no defectuós. Si ja existís al sistema l'alta hauria donat error i no hauria arribat a la part d'estadístiques, per això no es comprova si ja existeix a la taula *dim\_defectuos*. Seguidament es comprova si ja existeix al camp de control d'aquest mòdul a la taula de fets, si no es així, es crea amb 0% de components defectuosos com a valor inicial. Si existeix el valor a la taula de fets, s'actualitza amb el percentatge de components defectuosos per al codi del cotxe proporcionat trobats a la taula *dim\_defectuos*.

El procediment *baixa\_component* fa els mateix passos que el d'alta excepte comprovar que existeix, ja que si es vol donar de baixa un component es que ja existeix al sistema i a les taules estadístiques.

Es considera que un component és defectuós si proporciona un valor telemètric erroni. Per a representar això, es fa servir el disparador *chk\_telemetria* per a filtrar aquelles insercions definim com errònies (valors negatius, valor massa grans o massa petits, etc.). Si alguna d'aquestes condicions es dona al fer servir el procediment *alta\_telemetria*, el *handler* d'aquest procediment es farà càrrec d'actualitzar aquest component a la taula *dim\_defectuos* com a defectuós. Seguidament s'actualitzarà la taula de fets amb els nous valors.

#### **6.2.1.2 Mòdul 2** (Resultat de cada pilot per competició)

Per a obtenir la posició de cada pilot per competició es farà servir la taula *dim\_classificacio* ja que desa els punts que té cada pilot. El procediment *alta\_cursa* s'encarrega de, un cop realitzat l'insert d'alta, llegir la taula *dim\_classificacio* en ordre de punts per a la competició i temporada especificats a l'alta de cursa. Al mateix temps, es numeren les files obtingudes, i aquest valor serà el que representa la posició del pilot que requereix aquest mòdul. Seguidament, s'insereixen els valors de pilot, competició, temporada i posició a la taula de fets.

El procediment *baixa\_cursa* resta els punts que tenia aquest pilot per aquesta cursa de la taula *dim\_classificacio*, i seguidament actualitza la taula de fets amb els nous resultats.

#### **6.2.1.3 Mòdul 3** (Els 10 pilots amb més curses finalitzades)

El procediment *alta\_cursa* actualitza la taula *dim\_resultat* un cop realitzada l'alta per al pilot especificat als paràmetres. Si el pilot no es troba a la taula *dim\_resultat* i la posició final a la cursa és diferent de 0 (aquest valor s'utilitza per a especificar que el pilot no ha finalitzat la carrera), s'insereix el pilot a aquesta taula amb un 1 a curses finalitzades. Si existeix, es suma un 1 al valor que té actualment. Després de totes les condicions, s'actualitza la taula de fets amb els 10 primers pilots de la taula *dim\_resultat* ordenats per curses finalitzades.

El procediment *baixa\_cursa* restarà una unitat al nombre de curses finalitzades d'aquest pilot a la taula *dim\_resultat* si es va especificar a l'alta que aquest pilot va finalitzar aquesta cursa. Seguidament, s'actualitza la taula de fets amb la nova classificació de *dim\_resultat*.

#### **6.2.1.4 Mòdul 4** (Classificació per competició)

La implementació de la solució per a aquest mòdul és la mateixa que per al mòdul 2 però més senzilla. En el mòdul descrit prèviament es llegia la taula *dim\_classificacio* per a obtenir la posició dels pilots. En aquest cas només cal ordenar els resultats per ordre de punts i inserir aquests valors directament a la taula de fets.

#### **6.2.1.5 Mòdul 5** (Les 10 voltes més ràpides donades per circuit)

A la taula *dim\_volta\_rapida* es desen les curses i el temps per volta, i és el procediment *alta\_telemetria* i *baixa\_telemetria* els que s'encarreguen de mantenir-la actualitzada. A aquests dos procediments es considera el camp "àmbit" proporcionat a la crida al procediment. Si aquest és "temps per volta", es procedirà a actualitzar l'estadística d'aquest mòdul. Primer s'obté el codi del cotxe i el pilot a partir del codi del component especificat. També s'obté el circuit on s'està realitzant la cursa mitjançant la data en que es registra la dada telemètrica. Seguidament s'insereixen els atributs a la taula *dim\_volta\_rapida* i a continuació s'actualitza la taula de fets amb les 10 primeres files d'aquesta taula ordenades per temps per volta per al circuit en qüestió.



Mitjançant el procediment *baixa\_telemetria*, es considera novament el camp "àmbit" i si resulta ser "temps per volta" es procedeix a esborrar el registre d'aquesta volta de la taula *dim\_volta\_rapida*. Seguidament, s'actualitza la taula de fets de la mateixa manera que s'ha descrit a *alta\_telemetria*.

#### **6.2.1.6 Mòdul 6 (Temperatura més alta enregistrada)**

Al tractar-se només d'un valor únic, no es fa servir cap taula complementària per a tractar aquest mòdul. El procediment *alta\_telemetria* considera el camp "àmbit" i si resulta ser del tipus "temperatura", s'avaluarà si el valor és més gran que el desat a la taula de fets (si no existeix cap valor s'insereix directament). Simplement, si el valor trobat a la taula de fets és més petit, s'actualitza amb el nou valor. D'aquesta manera sempre es desarà la temperatura màxima trobada al sistema.

Quan es dona de baixa una dada telemètrica mitjançant *baixa\_telemetria* s'esborrarà el valor de temperatura màxima de la taula de fets i es substituirà amb la temperatura més alta enregistrada a la taula *telemetria*.

#### **6.2.1.7 Mòdul 7 (Els 5 patrocinadors que més diners aporten)**

Cada vegada que s'afegeix un nou patrocini al sistema mitjançant el procediment *alta\_patrocini* es dona d'alta aquest patrocini també a la taula *dim\_patrocini* si no ho estava prèviament. Si ja existeix, s'afegeix la quantitat del patrocini a la ja existent. Seguidament s'insereix a la taula de fets els 5 primers registres existents a la taula *dim\_patrocini* ordenats per la quantitat total aportada.

Quan es dona de baixa un patrocini, s'actualitzarà la taula *dim\_patrocini* restant al patrocini la quantitat que aquest patrocini representava. Seguidament, s'actualitza la taula de fets de la mateixa manera que a l'alta del patrocini.

#### **6.2.1.8 Mòdul 8 (Volum de dades telemètriques i equip que més dades ha enregistrat)**

Sempre que es dona d'alta una dada telemètrica mitjançant el procediment *alta\_telemetria*, es comprova si l'equip responsable del cotxe que ha generat la dada existeix a la taula *dim\_telemetria* per a aquesta temporada. Si no existeix, s'afegeix la temporada, l'equip i s'inicialitza el comptador de dades. Si ja existeix, se suma una unitat al comptador de dades totals. Seguidament, s'esborra de la taula de fets la entrada referent al volum acumulat de dades i la que fa referència a l'equip i la temporada actuals (si existeix) i s'insereix l'equip amb més dades generades que es troba per a aquesta temporada a la taula *dim\_telemetria*. També es s'insereix una entrada amb les dades totals generades aquell any obtinguda mitjançant una suma de les dades generades per tots els equip de la temporada en qüestió.

Quan s'executa el procediment *baixa\_telemetria*, es resta una unitat a la quantitat de dades generades per l'equip propietari del cotxe que va generar la dada (i referent a la temporada en

què es va generar) de la taula *dim\_telemetria*. Seguidament, s'actualitza la taula de fets de la manera especificada a l'alta.

#### **6.2.1.9 Mòdul 9** (Els 5 fabricants que més components aporten per competició)

Quan s'executa el procediment *alta\_component*, es comprova si el fabricant especificat es troba representat a la taula *dim\_component*. Si no es així, s'insereix el codi del fabricant i s'inicialitza el comptador de components d'aquest fabricant. Si ja existeix, es suma una unitat a la quantitat de components actual. A continuació, s'actualitza a la taula de fets les entrades referents a aquest mòdul (si existeixen) amb els 5 primers fabricants de la taula *dim\_component* ordenats per nombre de components.

Quan es dona de baixa un component mitjançant el procediment *baixa\_component* es fa el procediment contrari: es resta una unitat al comptador de components del fabricant en qüestió a la taula *dim\_component* i s'actualitza novament la taula de fets amb les 5 primeres entrades d'aquesta taula ordenades segons la quantitat de components enregistrats.

#### **6.2.1.10 Mòdul 10** (Els 10 pilots que més curses han guanyat)

El procediment que es fa servir per aquest mòdul es molt similar al descrit al mòdul 3 (ref. 6.2.1.3). El procediment *alta\_cursa* actualitza la taula *dim\_resultat*. Si el pilot no es troba a la taula *dim\_resultat* i el pilot ha guanyat la carrera (atribut posició és igual a 1), s'insereix el pilot a aquesta taula amb un 1 a curses guanyades. Si existeix, es suma un 1 al valor que té actualment. Després de totes les condicions, s'actualitza la taula de fets amb els 10 primers pilots de la taula *dim\_resultat* ordenats per curses guanyades.

El procediment *baixa\_cursa* restarà una unitat al nombre de curses guanyades d'aquest pilot a la taula *dim\_resultat* si es va especificar a l'alta que aquest pilot va guanyar aquesta cursa. Seguidament, s'actualitza la taula de fets amb la nova classificació de *dim\_resultat*.

#### **6.2.1.11 Mòdul 11** (Pilot amb la millor evolució)

Aquesta estadística no fa servir cap taula complementària, ja que totes les dades han d'estar disponibles a la taula de fets. Sempre que s'executa el procediment *alta\_cursa* es desa a la taula de fets la data de la cursa i els punts que té el pilot passat com a paràmetre per aquells data a partir de les dades actuals de la taula *dim\_classificació*. A continuació, es cerca a la taula de fets si aquest pilot ja te més dades referents a aquest mòdul estadístic, i es desa la evolució de punts entre l'estat actual i les curses prèvies, així com les dates actual i prèvia que s'ha avaluat. Finalment es disposarà de l'evolució del pilot entre totes les dates disponibles.

Quan s'executa el procediment *baixa\_cursa* s'esborren les referències de la taula de fet relacionades amb aquesta cursa i es resten els punts de les evolucions que van considerar aquesta cursa en el moment de la seva inserció.

Aquest mòdul té un prerequisit i es que les curses s'han d'inserir per ordre de data. Això és degut a que sempre es té en compte la evolució del pilot en el temps. Per tant, si s'insereixen les dates d'una cursa amb una data prèvia a les dades preexistents, la evolució no serà calculada correctament.

#### **6.2.1.12 Mòdul 12** (Cotxe que més carburant ha consumit per competició i temporada)

Quan s'executa el procediment *alta\_telemetria* amb l'atribut "àmbit" del tipus "consum", s'insereix a la taula *dim\_consum* (si no existeix prèviament) el cotxe, la temporada i la competició a la que pertany la dada. Si ja existeix l'entrada, s'actualitzarà amb el valor de consum especificat. Seguidament s'actualitzarà la taula de fets amb el cotxe amb més consum de carburant per a la temporada, cotxe i competició en qüestió trobat a la taula *dim\_consum*.

Quan s'executa el procediment *baixa\_telemetria* amb l'atribut "àmbit" del tipus "consum", es restarà el valor referent al consum de carburant d'aquesta dada del cotxe, temporada i competició corresponent localitzat a la taula *dim\_consum*. D'igual manera que al procediment d'alta, s'actualitzarà la taula de fets amb el nou valor màxim respecte al consum de carburant per a la temporada, cotxe i competició en qüestió trobat a la taula *dim\_consum*.

### **6.2.2 Actualització absoluta** (*funcionalitat addicional*)

S'han implementat una sèrie de procediments complementaris als requeriments de l'enunciat que actualitzen les estadístiques del sistema fent servir la informació disponible a les taules. Un cop executats, els procediments d'ABM poden executar-se amb normalitat per actualitzar les estadístiques en endavant.

No es descriurà amb detall el funcionament de cadascun d'ells, ja que a l'apartat anterior ja que la funcionalitat és molt semblant a la descrita a l'apartat anterior. Només afegir que, en aquest cas, s'afegeix la dificultat d'haver de recórrer totes les dades emmagatzemades mitjançant *loops* (algunes vegades es cerquen les dades mitjançant *loops* per temporada, altres per equip, per cursa, per competició, ...). Els procediments implementats són els següents:

- **Estadística\_classificació\_pilots:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_classificació* per donar resposta als mòduls 2 i 4 del repositori estadístic.
- **Estadística\_consum\_total:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_consum* per donar resposta al mòdul 12 del repositori estadístic.
- **Estadística\_curses\_finalitzades:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_resultat* per donar resposta al mòdul 3 del repositori estadístic.
- **Estadística\_curses\_resultat:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_resultat* per donar resposta al mòdul 10 del repositori estadístic.

- **Estadística\_patrocini\_màxim:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_patrocini* per donar resposta al mòdul 7 del repositori estadístic.
- **Estadística\_quantitat\_components:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_component* per donar resposta al mòdul 9 del repositori estadístic.
- **Estadística\_temperatura\_màxima:** Aquest procediment actualitza la taula de fets per donar resposta al mòdul 6 del repositori estadístic.
- **Estadística\_volta\_ràpida:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_volta\_rapida* per donar resposta al mòdul 5 del repositori estadístic.
- **Estadística\_volum\_telemetria:** Aquest procediment actualitza la taula de fets i la taula complementària *dim\_telemetria* per donar resposta al mòdul 8 del repositori estadístic.

### 6.2.3 Procediments d'execució

S'han creat una sèrie de procediments emmagatzemats (un per cada mòdul estadístic) la funció dels quals és executar les cerques estadístiques rebent com a paràmetres les dades especificades a l'enunciat. Els procediments són els següents:

- **me\_components\_defectuosos():** Cerca els percentatges de components defectuosos de cada cotxe (mòdul estadístic 1). No requereix cap paràmetre d'entrada.
- **me\_resultat\_pilots():** Cerca el resultat de cada pilot en cada competició disputada (mòdul estadístic 2). No requereix cap paràmetre d'entrada.
- **me\_carreres\_finalitzades():** Cerca els 10 pilots que han acabat més carreres (mòdul estadístic 3). No requereix cap paràmetre d'entrada.
- **me\_classificacio (competició, temporada):** Mostra la classificació de cada competició (mòdul estadístic 4). Cal especificar quina competició i temporada es vol mostrar.
- **me\_volta\_rapida(circuit):** Cerca les 10 voltes més ràpides donades a un circuit determinat (mòdul estadístic 5). Cal especificar el circuit que es vol mostrar.
- **me\_temperatura\_maxima():** Cerca la temperatura més alta enregistrada al sistema (mòdul estadístic 6). No requereix cap paràmetre d'entrada.
- **me\_top\_patrocinadors():** Mostra el TOP-5 de patrocinadors que aporten més diners als equips (mòdul estadístic 7). No requereix cap paràmetre d'entrada.
- **me\_dades\_telemetriques(temporada):** Mostra el volum de dades telemètriques enregistrades al sistema a un any concret i l'equip amb més dades enregistrades aquell any (mòdul estadístic 8). Cal especificar la temporada a la que es realitza la cerca.
- **me\_top\_fabricants():** Mostra el TOP-5 de fabricants que aporten més components (mòdul estadístic 9). No requereix cap paràmetre d'entrada.

- **me\_curses\_guanyades()**: Cercar els 10 pilots que han guanyat més carreres (mòdul estadístic 10). No requereix cap paràmetre d'entrada.
- **me\_millorevolucio(data d'inici, data final)**: Dades del pilot amb la millor evolució (mòdul estadístic 11). Cal especificar l'interval de dates en què es mesurarà la evolució.
- **me\_consum\_maxim(temporada, competició)**: Cerca quin és el cotxe que porta consumit més carburant per competició (mòdul estadístic 12). Cal especificar quina competició i temporada es vol mostrar.

## 7. Test

Per tal de provar la solució final el primer pas és inserir les dades, i això es pot fer de dues maneres. La primera de manera dinàmica:

1. Crear l'estructura de la base de dades (això significa, crear les taules, procediments i disparadors sense dades) executant l'arxiu annexa **CREAR\_DATABASE\_SIMPLE.sql**.
2. Inserir les dades fent servir els procediments d'alta executant l'arxiu annexa **INSERIR\_DADES.sql**.
3. Generar les dades telemètriques fent servir els procediments de simulació de l'arxiu annexa **INSERIR\_TELEMETRIA\_ALEATÒRIA.sql** (**Avís!** Aquest procediments **poden trigar gairebé una hora** en generar totes les dades, depèn de la potència de servidor).

La segona manera, més ràpida però que no fa servir els procediments implementats per a realitzar la inserció, és la manera estàtica:

1. Executar l'arxiu **CREAR\_DATABASE\_COMPLETA.sql** per tal de crear tota la base de dades, incloses les dades telemètriques (**Avís!** Mida del arxiu descomprimit > 200MB).

Aquesta és la descripció dels arxius esmentats:

- **CREAR\_DATABASE\_COMPLETA.sql**: aquest *script* conté totes les comandes necessàries per tal de crear la base de dades, les taules, els procediments emmagatzemats i els disparadors. S'ha d'executar en primer lloc per tal de poder procedir.
- **INSERIR\_DADES.sql**: aquest *script* no realitza inserts directament sinó que fa servir els procediments ABM implementats per tal d'omplir la base de dades. D'aquesta manera ja es prova de manera preliminar que aquests procediments funcionen correctament. S'ha d'executar en segon lloc per a poder procedir
- **INSERIR\_TELEMETRIA\_ALEATORIA.sql**: Per tal de generar les dades telemètriques que es fan servir al magatzem de dades (consum, volta ràpida i temperatura) es poden fer servir els procediments simuladors de dades per a obtenir dades noves cada cop. Aquests procediments (si s'executen tots ells) poden trigar uns 60 minuts, ja que insereixen gran quantitats de dades i actualitzen les estadístiques i *logs* a cada dada.

Per provar les funcionalitats un cop inserides les dades, s'han preparat uns arxius de proves que realitzen altes, baixes i modificacions tenint en compte les restriccions implementades i el tractament d'errors. Abans de començar és important comentar que s'ha intentat ordenar totes les comandes de tal manera que es puguin provar les funcionalitats amb certa facilitat. Als arxius 'TEST' però, les claus foranes poden referenciar entrades que encara no han estat creades o problemes d'aquest tipus. És per tant necessari tenir cura de que els arguments de

les comandes són correctes i que referencien els atributs correctes. També és important comentar que cal executar la comanda juntament amb el 'select' de la variable de retorn (RSP) per tal de poder veure el missatge de sortida. Es per això que cada comanda ve juntament amb aquesta variable.

Els arxius annexes necessaris per tal de reproduir les proves són els següents:

- **TEST\_PROCEDIMENTS\_ALTA.sql**: conté un conjunt de comandes que permeten provar cada procediment d'alta i el tractament d'errors en cada cas possible.
- **TEST\_PROCEDIMENTS\_BAIXA.sql**: conté un conjunt de comandes que permeten provar cada procediment de baixa i el tractament d'errors en cada cas possible.
- **TEST\_PROCEDIMENTS\_MODIFICACIÓ.sql**: conté un conjunt de comandes que permeten provar cada procediment de modificació i el tractament d'errors en cada cas possible.
- **TEST\_ PROCEDIMENTS ESTADÍSTICS**: conté un conjunt de crides a procediments per tal de provar el bon funcionament del repositori estadístic.

## 8. Conclusions

Aquest treball ha fet servir moltes de les eines disponibles a MySQL per a la utilització de base de dades. El meu coneixement era limitat a l'inici del treball en alguns àmbits (com per exemple, respecte al magatzem de dades), però ara ja conec en profunditat com crear procediments, disparadors, fer servir el control d'errors i dissenyar un magatzem de dades.

Els objectius s'han assolit en gran mesura, i la solució final proporciona una solució força sòlida i completa. És clar que el temps era molt ajustat donada la meva situació actual, però crec que he fet servir totes les eines i he après allò que calia aprendre. El resultat final, però, em deixa la sensació de que calia haver fet més proves, refinar més els requeriments de la tasca i haver cuidat més els petits detalls (que de ben segur d'alguns m'han he oblidat). Per exemple, els procediments de modificació no actualitzen el magatzem de dades, i vaig creure convenient aprofitar aquest temps per a enllestir altres parts del projecte ja que la tasca no consistia en fer res de nou, però requeria una gran quantitat de temps.

En quant a la planificació, estic força satisfet d'haver-la pogut seguir i d'haver controlat els riscos de manera adequada. El gran problema que he sofert ha estat la meva desconexió respecte al magatzem de dades. El projecte anava al dia amb la planificació, però arribat a aquest punt, vaig perdre gairebé dues setmanes intentant entendre la idea del repositori estadístic i, seguidament, implementant diverses solucions fins a trobar la que complia els requeriments de l'enunciat. Aquesta pèrdua de temps va ser desproporcionada respecte als dies per a incidències plantejats inicialment, així com el temps disponible a la planificació inicial per a realitzar el magatzem de dades. Aquest és el motiu principal d'haver deixat d'implementar algunes funcions i del poc refinament respecte a les proves i la presentació final del projecte.,

En resum, crec que els objectius personals s'han superat satisfactòriament. Sóc més àgil i tinc més coneixements sobre bases de dades i MySQL que quan vaig començar, això és el que em sembla més important. Tot i que sóc conscient, com e comentat, que la solució no compte amb el refinament final que un projecte d'aquestes característiques al món real hauria necessitat.



## 9. Glossari

**Screenecast:** en aquest cas s'utilitza per a referir-se a una gravació d'àudio juntament amb una presentació per a ser visualitzada de manera conjunta.

**Telemetria:** és una tecnologia que permet mesurar remotament magnituds físiques i enviar-les posteriorment per al seu anàlisi. En el nostre cas, les dades s'envien des dels cotxes i s'analitzen al magatzem de dades (veure terme en aquest glossari).

**PK, FK:** són les sigles en anglès de *Primary Key* i *Foreign Key* (en català, clau primària i clau forana). Serveixen per descriure, en el cas de la PK la clau única que referencia una fila d'una taula, i en el cas de la FK, atribut que referencia la PK d'una altra taula.

**PAC:** són les sigles de Prova d'Avaluació Continuada. Aquestes proves són comunament utilitzades a la UOC per a definir les entregues que es fan incrementalment durant el transcurs d'una assignatura (com és el cas d'aquest projecte).

**CRUD:** són les sigles en anglès per definir els procediments ABM (veure en aquest glossari).

**TFG:** són les sigles de Treball de Final de Grau, el propòsit d'aquest projecte.

**Disparador (*trigger*):** és un tècnica de programació que serveix per a executar comandes a la base de dades sense necessitat d'executar específicament una sentència SQL. EL codi s'executa sempre que es compleixen les condicions específiques per a executar-lo.

**GANTT (diagrama de):** és una eina gràfica que serveix per mostrar el temps dedicat a diferents tasques o activitats.

**UML:** són les sigles de *Unified Model Language* en anglès, un llenguatge gràfic que serveix per visualitzar, especificar i documentar un sistema (en aquest cas, una base de dades).

**LOG:** terme que, en aquest, fa referència a un registre de successos ocorreguts a un sistema informàtic (en aquest cas, una base de dades).

**FIA:** són les sigles de la Federació Internacional de Automobilisme, una organització de múltiples associacions automobilístiques coneguda per regular les competicions automobilístiques més importants del món.

**Metodologia en cascada:** és la manera d'organitzar les etapes del procés de desenvolupament de programari de tal manera que cada etapa comença un cop l'etapa anterior ha finalitzat.

**Magatzem de dades (*Data warehouse*):** conjunt de dades orientat a un àmbit concret que facilita la presa de decisions a una organització.

**Procediment emmagatzemat:** conjunt de comandes SQL que poden ser emmagatzemades a la base de dades;

**Procediments ABM:** Procediments emmagatzemats destinats a inserir, esborrar o modificar les taules de la base de dades.

**Script:** Arxiu simple de comandes. En aquest cas, de cerques en format SQL destinat a ser llegit per la base de dades.

**SQL:** és un llenguatge informàtic de consultes destinat a ser utilitzat per a cercar bases de dades.

**Repositori:** és un lloc centralitzat a un sistema informàtic on es desa informació d'algun tipus. En aquest cas, fa referència a la part de la base de dades on es desen les estadístiques.

## 10. Bibliografia

- [1] *Impacto de la nueva normativa de combustible en Formula 1*. URL: <http://planetadeporte.es/motor/formula-1/2014/impacto-normativa-combustible-635646>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [2] *Cicuito de Montmeló*. URL: <http://www.formulaf1.es/circuito/montmelo/>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [3] *Motores de formula 1*. URL: <http://www.soybalsport.com/image/swgpanell/cursos/28.pdf>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [4] *Federación Internacional del Automóvil (wiki)*: URL: [http://es.wikipedia.org/wiki/Federaci%C3%B3n\\_Internacional\\_del\\_Autom%C3%B3vil](http://es.wikipedia.org/wiki/Federaci%C3%B3n_Internacional_del_Autom%C3%B3vil)  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [5] *Pagina oficial de la FIA*: URL: [www.fia.com](http://www.fia.com)  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [6] *Sintaxi de MySQL*. URL: <http://dev.mysql.com/doc/refman/5.6/en/sql-syntax.html>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [7] "Data Warehousing by Example". Autor: *Barry Williams*. eBook de Amazon Digital Services, Inc. Numero ASIN: B00BE1Z130
- [8] *MySQL Error Handling in Stored Procedures*. URL: <http://www.mysqltutorial.org/mysql-error-handling-in-stored-procedures/>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [9] *Error Handling Examples*. URL: <http://www.devshed.com/c/a/mysql/error-handling-examples/>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [10] *Condition Handling*. URL: <http://dev.mysql.com/doc/refman/5.6/en/condition-handling.html>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [11] *SQLSTATE Codes*. URL: <http://www.csee.umbc.edu/portal/help/oracle8/server.815/a58231/appd.htm>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [12] *MySQL Stored Procedure Parameters*

<http://www.mysqltutorial.org/stored-procedures-parameters.aspx>  
(Visitat per ultima vegada al 10 de Juny de 2015)

- [13] *Using Triggers*. URL: <https://dev.mysql.com/doc/refman/5.5/en/triggers.html>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [14] *Validating data with triggers in MySQL*. URL:  
<http://cvuorinen.net/2013/05/validating-data-with-triggers-in-mysql/>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [15] *Almacén de datos*. URL: [http://es.wikipedia.org/wiki/Almac%C3%A9n\\_de\\_datos](http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos)  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [16] *Telemetría*. URL: <http://es.wikipedia.org/wiki/Telemetr%C3%ADa>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [17] *Consideraciones a tener en cuenta al diseñar una Base de Datos*. URL  
<https://firebird21.wordpress.com/2013/08/18/consideraciones-a-tener-en-cuenta-al-disenar-una-base-de-datos/>  
(Visitat per ultima vegada al 10 de Juny de 2015)
- [18] *Normalización de bases de datos*. URL  
[http://es.wikipedia.org/wiki/Normalizaci%C3%B3n\\_de\\_bases\\_de\\_datos](http://es.wikipedia.org/wiki/Normalizaci%C3%B3n_de_bases_de_datos)  
(Visitat per ultima vegada al 10 de Juny de 2015)