

Disseny i implementació d'una base de dades relacional per a un concessionari de vehicles

Cristóbal Duran Ballester
Enginyeria en Informàtica

Juan Martínez Bolaños
14 de Gener del 2013

Agraïments

El primer agraïment de tots el vull donar als meus pares, Joan i Maria, als quals dec que gràcies al seu esforç i recomanacions vaig arribar a poder estudiar a la Universitat de les Illes Balears, la qual cosa m'ha obert les portes a trobar el treball que he tingut aquests anys i que alhora m'ha permès estudiar el segon grau a la UOC.

El segon, i no per això menys important, a na Mònica, la qual ha tingut la paciència durant aquests tres any i mig de quedar caps de setmana i festius fora realitzar cap tipus d'activitat a causa de les PACs i/o exàmens que jo havia de realitzar, extensibles aquestes molèsties fins al punt d'haver de realitzar PACs estant de viatge.

Finalment a tots els meus grans amics, alguns *datilers* i d'altres que no, que en moments de moral baixa i amb ganes de deixar d'estudiar, m'han donat el sermó i m'han tornat a motivar per seguir fins al final per acabar amb aquest segon cicle.

Resum

Un gran concessionari de vehicles de primera i segona mà ens ha encarregat implementar un nou sistema que permeti controlar tant les vendes que es produeixen en diferents botigues com les reparacions i revisions que s'hi realitzin.

L'objectiu principal d'aquest projecte es aconseguir proporcionar una base de dades que permeti satisfer totes les necessitats del client. Per arribar a aconseguir-ho s'han de realitzar diferents etapes, les quals passen des d'acordar i definir els requeriments amb el client fins a certificar que el que s'ha implementat funciona correctament.

El primer que s'ha fet ha estat definir les funcions que ha de complir el sistema per satisfer els requeriments del client. Aquests requeriments són, entre d'altres, tenir desada la informació dels clients, la dels venedors, les vendes que s'han realitzat i la informació tant de les reparacions com les revisions que s'han realitzades o que s'han de realitzar.

En la segona fase s'ha definit un disseny (primer conceptual, després lògic i, finalment, físic) utilitzant diferents eines (com pot ser un model entitat-relació) per aconseguir definir el sistema que s'ha d'implementar.

Un cop definit el disseny físic, s'han implementat les diferents funcionalitats requerides pel client per a que funcionin en la versió *Oracle Database Express Edigion 11gR2*. Aquestes funcionalitats s'han comprovat i certificat utilitzant una sèrie d'entrades fictícies que han intentat reproduir diferents situacions reals a les quals es pot enfrontar el sistema.

Índex de continguts

1.-Introducció.....	8
1.1.-Requeriments del client.....	8
1.2.-Objectius.....	8
1.3.-Enfocament i metodologia utilitzada.....	9
1.4.-Planificació.....	10
1.4.1.-Tasques i temps dedicat.....	10
1.4.2.-Diagrama de Gantt.....	12
1.4.2.1.-Detall fases.....	12
1.4.2.1.1.-Planificació (PAC1).....	12
1.4.2.1.2.-Anàlisi i disseny (PAC2).....	13
1.4.2.1.3.-Implementació i proves (PAC3).....	13
1.4.2.1.4.-Lliurament Final.....	13
1.5.-Recursos necessaris.....	14
1.5.1.-Maquinari.....	14
1.5.2.-Programari.....	14
1.5.3.-Recursos tècnics.....	14
1.5.4.-Recursos humans.....	15
1.6.-Anàlisis dels riscos i pla de contingència.....	15
1.7.-Productes obtinguts.....	16
1.8.-Breu descripció dels altres capítols de la memòria.....	16
2.-Definició i anàlisi de requisits.....	17
2.1.-Regles de Negoci.....	17
2.2.-Requisits funcionals.....	18
3.-Disseny.....	23
3.1.-Disseny Conceptual.....	23
3.1.1.- Casos d'ús.....	23
3.1.2.- Diagrama entitat-relació en UML.....	27
3.1.2.1.-Enunciat.....	27
3.1.2.2.-Mòdul Datawarehouse.....	27
3.1.2.3.-Mòdul Control.....	28
3.1.3.- Entitats i atributs.....	29
3.1.3.1.- Enunciat.....	29
3.1.3.2.- Mòdul Datawarehouse.....	30
3.1.3.2.1.- Taules de Dimensions.....	30
3.1.3.2.2.- Taules de Fets.....	30
3.1.3.3.- Mòdul de control.....	30
3.1.4.- Relacions.....	31
3.2.- Disseny lògic.....	32
3.2.1.- Model relacional.....	32
3.2.1.1.- Enunciat.....	32
3.2.1.2.- Mòdul Datawarehouse.....	34
3.2.1.2.1.- Taules de Dimensions.....	34
3.2.1.2.2.- Taules de Fets.....	34
3.3.- Disseny físic.....	35

3.3.1.- Creació de la BD.....	35
3.3.2.- Creació del tablespace.....	35
3.3.3.- Creació d'usuari.....	36
3.3.4.- Creació de taules.....	36
3.3.5.- Creació d'índexs	36
3.3.6.- Creació de seqüències.....	36
4.- Implementació.....	37
4.1.-Package Control.....	37
4.2.-Package Client.....	41
4.3.-Package Cotxes.....	42
4.4.-Package DWH.....	47
4.5.-Package RRHH.....	52
4.6.-Package Taller.....	55
4.7.-Package Vendes.....	60
4.8.-Package Consultes.....	64
5.-Proves.....	67
5.1.-Càrrega inicial de dades.....	67
5.2.-Procediments ABM.....	67
5.3.-Consultes.....	68
5.3.1.-Client per DNI o CIF.....	68
5.3.2.-Client per ID.....	68
5.3.3.-Venedor per ID.....	68
5.3.4.-Model per ID.....	69
5.3.5.-Cotxe per ID.....	69
5.3.6.-Llistat de vendes.....	69
5.3.7.-Llistat de revisions.....	70
5.3.8.-Llistat de reparacions.....	70
5.3.9.-Avisar revisions per data.....	71
5.4.-Mòdul Datawarehouse.....	72
5.4.1.-Inicialització dimensió data.....	72
5.4.2.-Càrrega dimensions i fets.....	72
5.4.3.-Temps reparació cotxe.....	72
5.4.4.-Rànquing venedor.....	72
5.4.5.-Reparacions per mes.....	72
5.4.6.-Revisions per mes.....	73
5.4.7.-Vendes per mes.....	73
5.4.8.-Despesa per client.....	73
5.5.-Mode Test.....	74
5.5.1.-Recompte dels registres per taula.....	74
5.5.3.-Càrrega, modificació i esborrada de dades	74
5.5.4.-Recompte dels registres per taula.....	74
5.6.-Logs.....	75
5.6.1.-Cerca d'errors.....	75
5.6.2.-Provocar error.....	75
5.6.3.-Desactivar LOG i provocar error.....	75
5.6.4.-Consultar LOG per ID.....	76
5.6.5.-Consultar LOG per rang de temps.....	76

6.-Valoració econòmica.....	77
7.-Conclusions.....	77
8.-Glossari.....	78
9.-Bibliografia.....	78
9.1.-Documentació.....	78
9.2.-Enllaços d'internet.....	78
10.-Annex.....	79

Índex de figures

1: Metodologia.....	9
2: Taula distribució hores.....	10
3: Diagrama de Gantt Global.....	12
4: Diagrama de Gantt PAC1.....	12
5: Diagrama de Gantt PAC2.....	13
6: Diagrama de Gantt PAC3.....	13
7: Diagrama de Gantt Lliurament Final.....	13
8: Casos d'ús DWH i Cotxes.....	24
9: Casos d'ús Vendes, Client i Taller.....	25
10: Casos d'ús RRHH i Control.....	25
11: Casos d'ús Consultes.....	26
12: Diagrama E/R Enunciat.....	27
13: Diagrama E/R Datawarehouse.....	27
14: Diagrama E/R Mòdul de Control.....	28
15: Valoració econòmica.....	77

1.-Introducció

1.1.-Requeriments del client

Un gran concessionari de vehicles de primera i segona mà vol incorporar un sistema per controlar les vendes i reparacions que es produeixen a les diferents botigues.

Necessita una BD que permeti desar tant la informació relacionada amb les vendes (característiques del vehicles, extres, informació del venedor, etc...) com la informació de les reparacions i les revisions que es duguin a terme en els tallers (client, motiu de la reparació, quant de temps s'ha trigat en solucionar-la, etc..). El client vol també processar tota aquesta informació mitjançant una sèrie de procediments per a poder obtenir tant estadístiques com llistats interessants per al funcionament del negoci.

A més, es vol que la BD sigui escalable per facilitar la incorporació de noves necessitats que sorgeixin en el futur.

1.2.-Objectius

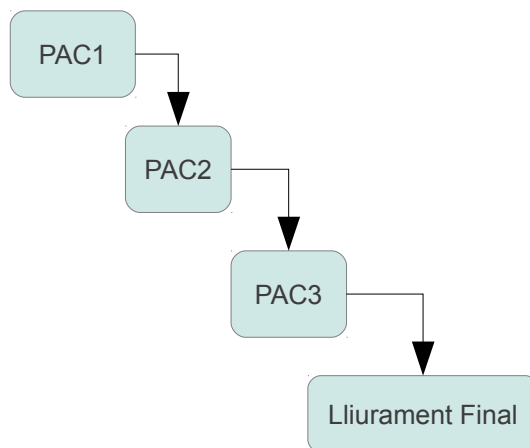
Els objectius del projecte que s'extreuen dels requeriments que ha indicat el client són els següents:

- Emmagatzemament de les dades relacionades amb les de vendes de cotxes:
 - Dades de les botigues
 - Dades dels cotxes
 - Dades dels venedors
 - Dades dels clients
- Emmagatzemament de les dades relacionades amb les reparacions i revisions de cotxes:
 - Dades dels tallers
 - Dades de les reparacions
 - Dades de les revisions (pot ser que en una revisió no es repari res)
 - Dades dels mecànics
- Magatzem de dades (*datawarehouse*) que obtenir estadístiques de les dades emmagatzemades en les taules, com poden ser:
 - Temps mitjà que es triga en reparar un cotxe
 - Llistat del millors venedors
- Procediments i triggers:
 - Procediments i triggers per a la gestió de les dades
- Dades:
 - Dades per a la inicialització de la BD
 - Joc de proves per certificar el correcte funcionament de tot el model implementat

1.3.-Enfocament i metodologia utilitzada

La metodologia que s'utilitzarà per al cicle de vida d'aquest projecte serà el *model en cascada*, el qual ordena les etapes del cicle de vida del programari de forma que per a inicialitzar una nova fase primer s'ha d'haver acabat l'anterior.

Al constar el projecte de quatre entregues clau, la representació del cicle de vida en cascada quedaria de la següent forma:



1: Metodologia

Les fases que utilitzarem en aquest projecte aplicant aquesta metodologia són:

- Recollida i anàlisi de requeriments: es realitza un estudi dels requeriments que necessita el client i es determina l'abast del projecte, amb la corresponent planificació.
- Disseny: es defineix una solució als requeriments obtinguts en la fase anterior i es defineixen les estructures necessàries.
- Implementació i proves: es desenvolupen les taules, els procediments, claus primàries, etc... que permetran que la BD compleixi les funcionalitats requerides. Amb els diferents jocs de proves es comproven que totes aquestes funcionalitats siguin correctes.
- Lliurament Final: es prepara el lliurament final del producte al client.

1.4.-Planificació

Des de la data inicial del projecte fins a la data final hi ha unes 17 setmanes. Per a distribuir les diferents entregues i el temps disponible a cada una d'elles s'han tingut en compte tant els caps de setmanes que hi ha entre cada una d'elles com els festius.

Donat que els caps de setmana i els festius és quan més hores es poden dedicar a aquest projecte, s'ha considerat que aquests dies es poden dedicar 6 hores al dia (es deixa marge per si qualcun d'aquests dies s'han de fer més de 6 hores) i els dies entre setmana sols mitja hora al dia.

D'aquesta forma les hores dedicades a cada entrega del projecte (seguint la planificació proposada) queden distribuïdes de la següent forma:

Inici	Entrega	Dies	CapSet	Festius	Hores	Activitat	Descripció
19/09/12	08/10/12	20	3	0	43	PAC1	Pla de treball
09/10/12	12/11/12	35	5	2	83.5	PAC2	Anàlisi i disseny
13/11/12	13/12/12	31	4	1	65	PAC3	Implementació i proves
14/12/12	14/01/13	32	5	3	87.5	Lliurament Final	Memòria, treball i presentació
21/01/13	25/01/13	5	0	0	-	Debat	Debat virtual

2: Taula distribució hores

El total d'hores dedicades al projecte s'estimen que seran 279 hores, el que equival a unes 16.41 hores de mitja a la setmana.

1.4.1.-Tasques i temps dedicat

Les tasques que consider que s'haurien de realitzar i la durada que crec que tindran són:

Planificació (43 hores):

- Lectura de l'enunciat: *0.5 hores*
- Identificació i definició dels objectius: *13 hores*
- Definició de l'enfocament i metodologia utilitzada: *8.5 hores*
- Identificació de tasques a realitzar: *6 hores*
- Redacció de la PAC1: *8.5 hores*
- Revisió de la PAC1: *6 hores*
- Lliurament de la PAC1: *0.5 hores*

Anàlisi i disseny (83.5 hores):

- Revisió de requisits:
 - Definició detallada dels requisits inicials: *6.5 hores*
 - Actualització dels requisits: *7 hores*
- Disseny conceptual
 - Identificació d'entitats de l'enunciat: *6 hores*
 - Identificació d'entitats del mòdul de Datawarehouse: *3 hores*
 - Identificació d'entitats del registre d'accions (log): *3 hores*
- Disseny lògic: *14.5 hores*
- Disseny físic: *14.5 hores*
- Instal·lació del Oracle Express 11gR2: *8 hores*
- Instal·lació del SQLDeveloper: *3 hores*
- Recerca informació de PL/SQL: *9 hores*
- Revisió de la PAC2: *8.5 hores*
- Lliurament de la PAC2: *0.5 hores*

Implementació i proves (65 hores):

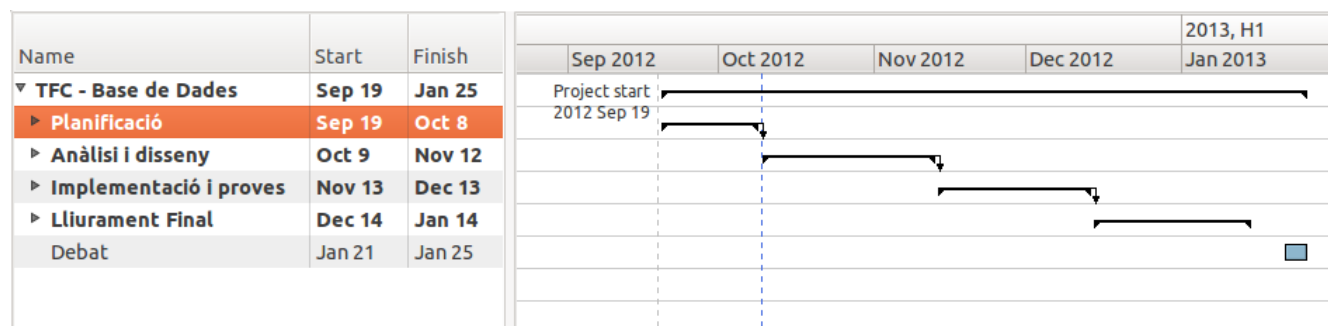
- Creació d'esquemes i taules: *14 hores*
- Creació dels procediments de consulta: *15 hores*
- Implementació Datawarehouse: *6 hores*
- Implementació registre (log): *4 hores*
- Elaboració i execució dels jocs de proves: *14 hores*
- Redacció de la PAC3: *8 hores*
- Revisió de la PAC3: *3.5 hores*
- Lliurament de la PAC3: *0.5 hores*

Lliurament final:

- Execució proves finals: *27.5 hores*
- Redacció de la Memòria: *33 hores*
- Creació de la Presentació: *14 hores*
- Preparació del treball pràctic: *12.5 hores*
- Lliurament Final: *0.5 hores*

1.4.2.-Diagrama de Gantt

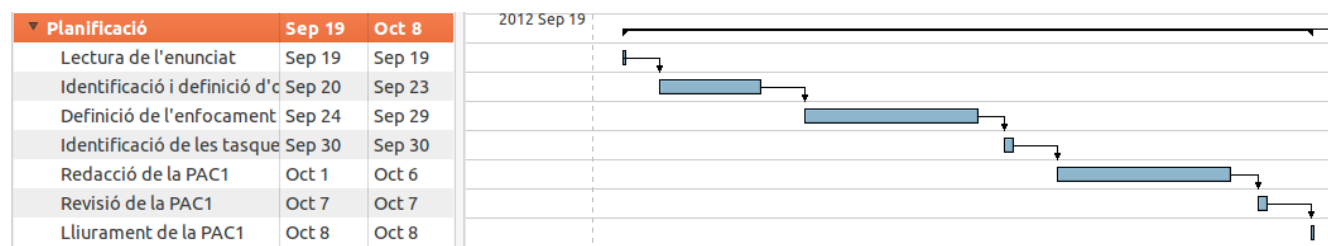
Per a realitzar el diagrama de *Gantt* s'ha utilitzat el programari *planner*. La vista general del diagrama queda de la següent forma:



3: Diagrama de Gantt Global

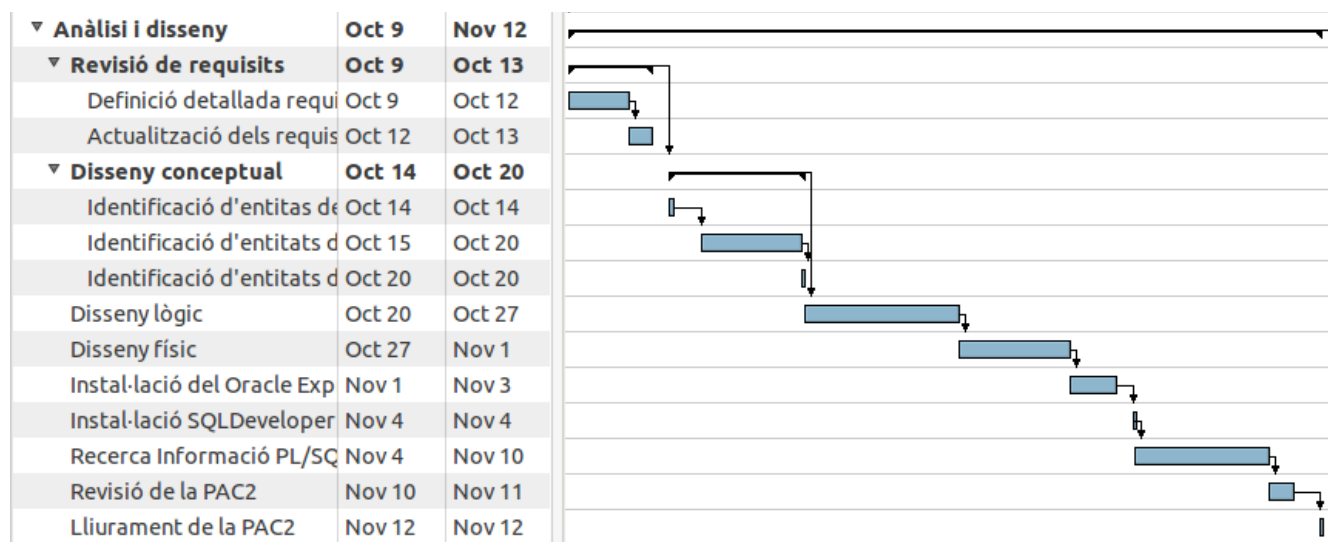
1.4.2.1.-Detall fases

1.4.2.1.1.-Planificació (PAC1)



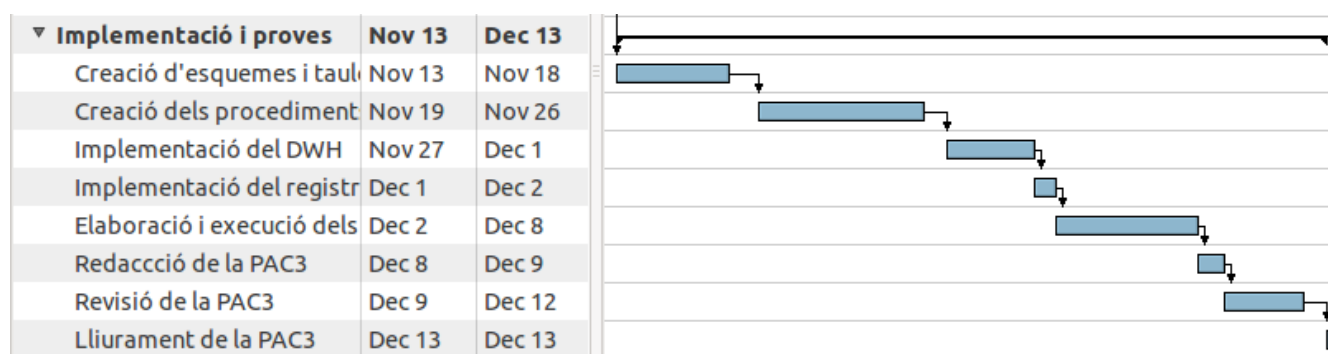
4: Diagrama de Gantt PAC1

1.4.2.1.2.-Anàlisi i disseny (PAC2)



5: Diagrama de Gantt PAC2

1.4.2.1.3.-Implementació i proves (PAC3)



6: Diagrama de Gantt PAC3

1.4.2.1.4.-Lliurament Final



7: Diagrama de Gantt Lliurament Final

1.5.-Recursos necessaris

1.5.1.-Maquinari

El maquinari que s'utilitzarà tant per a executar el SGBD com el resta d'aplicacions serà el mateix, és a dir, es desenvoluparà de forma local.

Les característiques principals del *host* són:

- 4GB de RAM
- Processador AMD Quad-Core
- Disc Dur Serial ATA2 de 1TB de capacitat.

1.5.2.-Programari

El sistema operatiu que s'utilitzarà és un UBUNTU 12.04 de 64bits el qual inclou, entre altre programari, el *planner* i el *libreoffice*.

Sobre el sistema operatiu anterior s'executarà el SGDB ORACLE Express 11gR2, el qual és una versió del famós SGBD que permet utilitzar fins a 11GB d'emmagatzemament en disc, 1 GB de RAM i una CPU.

Per a connectar a la BD i fer tant consultes, com crear les taules, procediments, triggers, indexes, etc... s'utilitzarà el *SQLDeveloper*, la qual és una eina programada amb *java* (i, per tant, multiplataforma) pròpia d'ORACLE que ofereix les funcionalitats necessàries per dur a terme el projecte de forma ràpida i eficient.

1.5.3.-Recursos tècnics

Els recursos tècnics són tots aquells recursos que permetran a un recurs humà desenvolupar de forma adequada les seves funcions.

El model de metodologia utilitzat en aquest projecte ens dona les fases clau per a la construcció del software que complesqui els requisits indicats pel client. Aquestes fases són:

- **Anàlisi i disseny:** s'extreuen del client les necessitats del projecte i es dissenya l'estructura que les satisfarà. S'utilitzen diagrames de flux i diagrames de classes, entre altres recursos tècnics.
- **Implementació:** es programa el disseny obtingut en el punt anterior. En aquest projecte s'utilitzen els llenguatges de programació PL/SQL i SQL.
- **Execució de proves:** es realitzen proves unitàries (centralitzades en una part molt concreta del codi programat) mitjançant SQL per detectar els possibles errors del programa.
- **Correcció d'errors:** en cas de detectar-se errors, s'han de tornar a implementar el procediments involucrats per tal de solucionar-los.
- **Documentació:** s'ha de documentar tota la informació relacionada amb el projecte.

1.5.4.-Recursos humans

Els recursos humans són el conjunt de treballadors que participen en la realització del projecte. S'estima que hi haurà d'haver uns cinc perfils de treballadors, depenent de la funció que realitzi:

- **Cap de projecte:** s'encarrega de planificar de forma detallada el projecte, d'organitzar l'equip que hi participa, d'administrar els costos i el temps d'entrega, d'assignar les responsabilitats, de prendre les decisions oportunes, de comunicar-se amb el client i d'anticipar-se als possibles problemes.
- **Analista:** és l'encarregat de definir en el disseny les solucions a les necessitats del client.
- **Programador:** és l'encarregat d'implementar el disseny proporcionat per l'analista.
- **Tester (comprovar):** s'encarrega d'executar proves en el sistema implementat per avaluar el correcte funcionament del que s'ha o s'està implementant.
- **Documentalista:** elabora la documentació que engloba tota la informació del projecte.

1.6.-Anàlisis dels riscos i pla de contingència

Es recomanable enumerar les diferents situacions de risc que es poden donar al llarg de l'execució del projecte, així com també s'han de definir per a cada una d'elles les mesures preventives adoptades per fer-li front.

- **Riscos de costos**
 - *Causa:* augment dels costos de desenvolupament previstos per una estimació incorrecte durant la fase de planificació.
 - *Mesura preventiva:* s'ha de fer una previsió de costos més elevada per evitar una desviació desmesurada entre el cost estimat i el cost real.
- **Riscos de calendari**
 - *Causa:* sobrepassar el calendari previst per qualsevol motiu, com pot ser un increment dels esforços en la solució de qualque punt del projecte, una mala estimació de temps, una pèrdua de recursos humans no prevista o una assignació de recursos no planejada.
 - *Mesura preventiva:* a la planificació s'ha de tenir en compte els factors anteriors i s'ha de definir tenint en compte el pitjor dels casos.
- **Riscos de dades**
 - *Causa:* pèrdua de dades per qualque tipus d'anomalia (esborrat accidental o problemes de disc dur).
 - *Mesura preventiva:* realitzar una còpia de seguretat periòdica en un segon dispositiu d'emmagatzemament diferent al de treball. Un exemple pot ser el *dropbox*, eina que permet desar en el núvol qualsevol fitxer.

- **Riscos de maquinari**
 - *Causa*: problemes amb el maquinari de treball que impossibilita continuar la realització del projecte en el mateix.
 - *Mesura preventiva*: disposar d'un maquinari secundari sobre el qual es pugui recuperar la còpia de seguretat periòdica i continuar el desenvolupament normal del projecte (assumint una pèrdua dels darrers canvis no desats en el dispositiu secundari).

1.7.-Productes obtinguts

Els productes obtinguts un cop s'hagi finalitzat el projecte seran els següents:

- **Producte**: inclou els scripts necessaris per a la creació de la BD, l'usuari de connexió, les taules, etc...
- **Memòria**: document amb tota la informació referent al projecte: planificació, requisits, resultats, etc...
- **Presentació**: resum clar i concís en diapositives on s'explica el treball realitzat.

1.8.-Breu descripció dels altres capítols de la memòria

- **2.-Definició i anàlisi de requisits**: es defineixen i es s'analitzen amb el client els requeriments que ha de complir el sistema per satisfer les seves necessitats.
- **3.-Disseny**: a partir dels requeriments del client es realitza un disseny de la base de dades a implementar.
- **4.-Implementació**: s'implementa cada una de les funcionalitats que requereix el sistema.
- **5.-Proves**: es realitzen una sèrie de proves per a comprovar el correcte funcionament del sistema i verificar que es satisfan les necessitats del client.
- **6.-Valoració econòmica**: valoració del cost del producte a desenvolupar en base al nombre d'hores que hi participarà cada un dels diferents perfils.
- **7.-Conclusions**: valoracions finals del projecte realitzat.
- **8.-Glossari**: definició de termes utilitzats en aquest document.
- **9.-Bibliografia**: recursos utilitzats durant la realització del projecte.

2.-Definició i anàlisi de requisits

En aquest punt és defineixen i s'analitzen tant les regles de negoci com els requisits funcionals que ha de complir el sistema.

2.1.-Regles de Negoci

Les regles de negoci que s'extreuen de les consultes realitzades amb el client són les següents:

Identificador	Descripció
RN1	<p>Es determina que les opcions de compra d'un vehicle que ha de contemplar el sistema són:</p> <ol style="list-style-type: none">1. Un vehicle nou que s'ha de reservar a la fàbrica: el client pot configurar a la carta les diferents opcions (amb el seu corresponent preu) i triar el model que vol. El que es fa és realitzar una reserva a la fàbrica. Un cop es rep el vehicle es dóna d'alta en el sistema i, quan s'entrega al client, es crea la corresponent venda.2. Un vehicle nou que es troba en estoc: el vehicle ja es troba al concessionari i té unes opcions incloses (amb el seu corresponent preu) que no poden variar. L'avantatge pel client és que tindrà disposició pràcticament immediata del vehicle.3. Un vehicle de segona mà: quan s'insereix en el sistema un vehicle de segona mà per vendre, qualcú (un venedor o taxador) determinarà el preu segons el seu estat i els extres que incorpora.
RN2	<p>Els cotxes sobre els quals es poden realitzar revisions i reparacions poden tenir tres “procedències” diferents:</p> <ol style="list-style-type: none">1. Cotxe nou comprat al concessionari2. Cotxe de segona mà comprat al concessionari3. Cotxe no comprat al concessionari
RN3	<p>Sobre les revisions:</p> <ol style="list-style-type: none">1. Les revisions s'inseriran amb la data de quan s'han de realitzar. Hi haurà un procediment diari que s'encarregarà d'enviar un SMS una setmana abans recordant la cita corresponent.2. El preu d'una revisió es calcularà multiplicant les hores dedicades pels diferents mecànics pel preu de l'hora corresponent a la seva categoria.

RN4	<p>Sobre les reparacions:</p> <ol style="list-style-type: none"> 1. Pot venir un client directament amb el vehicle espenyat que no l'ha comprat en el concessionari. Per tant, s'ha de poder donar d'alta vehicles per poder reparar-los que no s'han comprat en el concessionari. 2. El preu d'una averia es calcularà multiplicant les hores dedicades pels diferents mecànics pel preu de l'hora corresponent a la seva categoria més el preu de les peces que s'han canviat. 3. Donat que ens trobam en temps de crisi, es veu recomanable que el sistema permeti realitzar un pressupost de reparació, per així orientar al client el preu que tindrà abans de realitzar-la.
-----	--

2.2.-Requisits funcionals

Els requisits funcionals que s'extreuen de les consultes realitzades amb el client són les següents:

Identificador	Descripció
RF1	<p>S'han de guardar les reserves d'un vehicle nou que s'ha comanat a fàbrica. Un cop es rebí el vehicle, es desaran la resta de dades del cotxe (bastidor, etc). D'una reserva, s'ha de desar:</p> <ul style="list-style-type: none"> • Client que ha realitzat la reserva. • Marca del cotxe reservat • Model del cotxe reservat • Opcions del cotxe reservat • Preu total del cotxe reservat <p>Un cop s'ha rebut el cotxe, una reserva passarà a ser una venda i es desarà les dades del vehicle.</p>
RF2	<p>D'un vehicle es guardaran les següents dades:</p> <ul style="list-style-type: none"> • Codi del vehicle, que permetrà identificar inequívocament un vehicle. Serà un atribut obligatori.

	<ul style="list-style-type: none"> • Propietari del vehicle. Si és un cotxe en venda, el propietari serà el concessionari. • Número de bastidor del vehicle, per identificar un vehicle. No s'utilitza la matrícula, ja que un cotxe nou pot no estar matriculat. • Matrícula del vehicle. No serà un camp obligatori, però es desarà un cop es sàpiga. • Marca del vehicle, per determinar la marca que ha fabricat el vehicle. • Model del vehicle, per determinar quin model de vehicle és i el preu que té. Amb el model s'inclouen les característiques bàsiques del vehicle, com poden ser: si funciona amb electricitat/benzina/gasol, CV de potència, cilindrada, número de portes, etc... <p>Si es tracten de cotxes per vendre, s'ha de poder guardar la següent informació:</p> <ul style="list-style-type: none"> • Data d'entrada del cotxe en el concessionari. Per així controlar quan de temps fa que es ven el cotxe. • Opcions del vehicle. Per saber, per exemple, les opcions que inclouen els cotxes que es venen, tant de segona mà com nous que hi ha en estoc. <p>Si, a més, es tracta d'un cotxe de segona mà, s'ha de poder desar el preu que es demana pel vehicle, el qual haurà estat posat per un venedor o un taxador.</p>
RF3	<p>El model ha de permetre guardar les dades relacionades amb el client:</p> <ul style="list-style-type: none"> • El codi de client, que s'utilitzarà com a identificador únic del client. Es podria utilitzar el DNI, però és decideix utilitzar el codi pel fet que pot ser que el client sigui una empresa. • El DNI/CIF del client. • Nom del client. Si és una empresa serà aquí on es desarà el nom de l'empresa. • Primer llinatge del client. Si és una empresa es deixarà buit. • Segon llinatge del client. Si és una empresa o un client estranger, es deixarà buit. • Telèfon de contacte. S'utilitzarà per posar-se en contacte amb el client i per enviar el avisos de revisions.

RF4	<p>El model ha de permetre guardar les dades relacionades amb el venedor:</p> <ul style="list-style-type: none"> • El codi del venedor, que s'utilitzarà com a identificador únic del venedor. Es podria utilitzar el DNI, però és decideix utilitzar el codi. • Nom del venedor. • Primer llinatge del venedor. • Segon llinatge del venedor. Si és estranger, es deixarà buit. • Telèfon del venedor
RF5	<p>El model ha de permetre guardar les dades necessàries per gestionar la venda d'un cotxe. Aquestes com a mínim són:</p> <ul style="list-style-type: none"> • Data de la venda, per saber quan es va realitzar la venda • Número del bastidor del cotxe. Permet identificar inequívocament el vehicle que s'ha venut • Client que ha comprat el cotxe. • Venedor que ha venut el cotxe. • Extres que inclou el cotxe. Aquests extrems poden ser: aire condicionat, ABS, tancament centralitzat, etc... • Preu que s'ha pagat pel cotxe. Preu definitiu que s'ha pagat pel vehicle. En el cas dels vehicles nous inclou el preu dels extrems.
RF6	<p>El model ha de permetre guardar les dades necessàries per gestionar les reparacions d'un cotxe. Aquestes com a mínim són:</p> <ul style="list-style-type: none"> • Data d'inici de la reparació. Data que arriba el cotxe al taller. • Data fi de la reparació. Data en que es dona com a finalitzada la reparació i s'avisat al client perquè el vengui a cercar. • Cotxe al qual es fa la reparació. • Motiu de la reparació • Mecànic que ha participat en una revisió, hores que hi ha dedicat i el preu de cada una de les hores. El preu de les hores depenen de la categoria del mecànic: planxista, mecànic, electricista, etc...

RF7	<p>El model ha de permetre guardar les dades necessàries per gestionar les revisions d'un cotxe. Aquestes com a mínim són:</p> <ul style="list-style-type: none"> • Data de la revisió • Cotxe que ha de fer la revisió • Motiu de la revisió • Mecànic que ha participat en una revisió, hores que hi ha dedicat i el preu de cada una de les hores. El preu de les hores depenen de la categoria del mecànic: planxista, mecànic, electricista, etc...
RF8	<p>El sistema ha de permetre, com a mínim, realitzar les funcionalitats següents:</p> <ul style="list-style-type: none"> • Procediments d'ABM (Alta + Baixa + Modificació) de cotxes i la informació relacionada: extres, model, marca, etc... • Procediments d'ABM de reserves. • Procediments d'ABM de vendes. • Procediments d'ABM de reparacions. • Procediments d'ABM de revisions. • Procediments d'ABM de clients. • Procediments d'ABM de venedors. • Procediments d'ABM de mecànics i les categories corresponents. • Procediments de consulta que permetin obtenir: <ol style="list-style-type: none"> 1. Donat un client i una operació (reserva, venda, reparació, revisió) passats paràmetre, retornarà el llistat de totes les operacions del tipus sol·licitat que ha realitzat el client en el concessionari. 2. Donat un venedor i una operació de venda o reserva, retornarà el llistat de totes les operacions del tipus sol·licitat que hagi realitzat el venedor. 3. Donat un mecànic i una operació de revisió o reparació, retornarà el llistat de totes les operacions del tipus sol·licitat en les que ha participat el mecànic, el temps que hi ha dedicat i el cost que tenia cada hora quan hi va participar.
RF9	<p>S'ha d'implementar un magatzem de dades que permeti extreure estadístiques que indiquin les següents dades:</p> <ol style="list-style-type: none"> 1. El temps mitjà que es triga en reparar un cotxe. Cada dia es revisaran totes les reparacions finalitzades i es mirarà l'hora d'entrada al taller del vehicle i l'hora que s'ha donat per finalitzada la reparació.

	<ol style="list-style-type: none"> 2. Quins són els millors venedors. Cada dia es revisaran totes les vendes realitzades per a cada un dels venedors. 3. Quins mesos es realitzen més revisions. A principi de mes es revisaran totes les revisions que s'han realitzat al més anterior. 4. Quins mesos es realitzen més reparacions. A principi de mes es revisaran totes les reparacions que s'han realitzat al més anterior. 5. Quins mesos es realitzen més vendes. A principi de mes es revisaran totes les vendes que s'han realitzat al més anterior. 6. Quins són els clients que han gastat més doblers. Cada dia a la nit es carregaran en una part del datawarehouse totes les operacions que provoquen ingressos (venda, reparació i revisió) i el client que les ha realitzat. Així es tindrà un rànquing dels millors clients del concessionari. <p>Es decideix que s'executaran processos nocturns que actualitzaran diàriament les dades del datawarehouse. Es decideix que siguin nocturns perquè és quan els usuaris no utilitzaran el sistema i així la càrrega addicional no els hi afectarà.</p>
RF10	<p>Hi haurà un mòdul de control que permetrà:</p> <ol style="list-style-type: none"> 1. Definir si el sistema es troba en mode “prova”. Si és així, cap dels canvis que es realitzin es desaran. Servirà per poder realitzar proves. 2. Desar totes les accions que s'hagin realitzat al sistema. 3. Desar tots els errors que s'hagin produït en el sistema.

3.-Disseny

3.1.-Disseny Conceptual

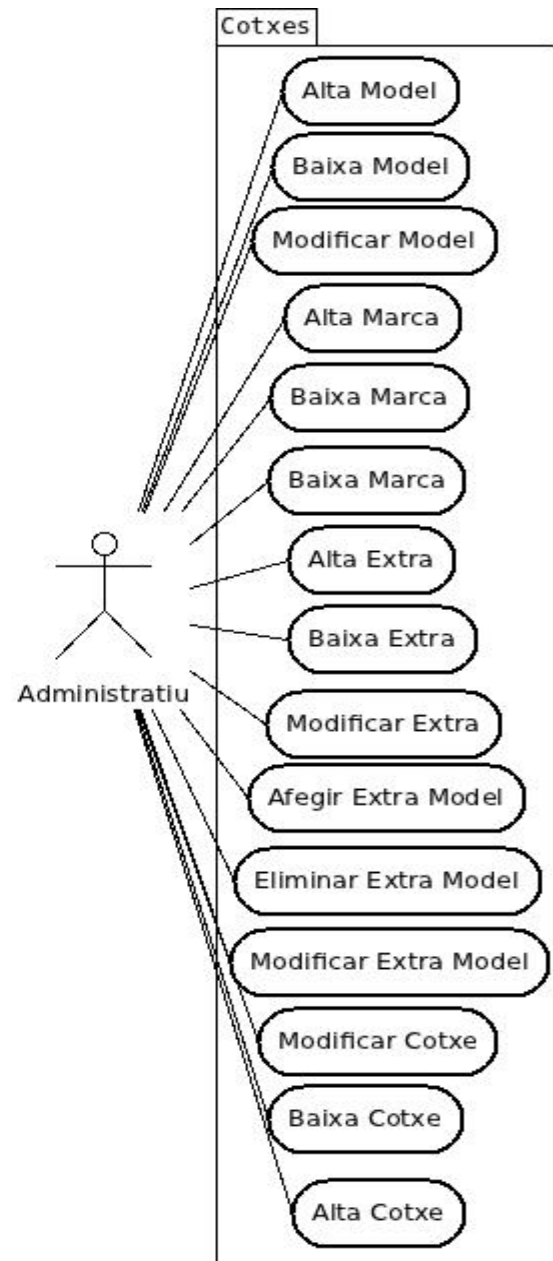
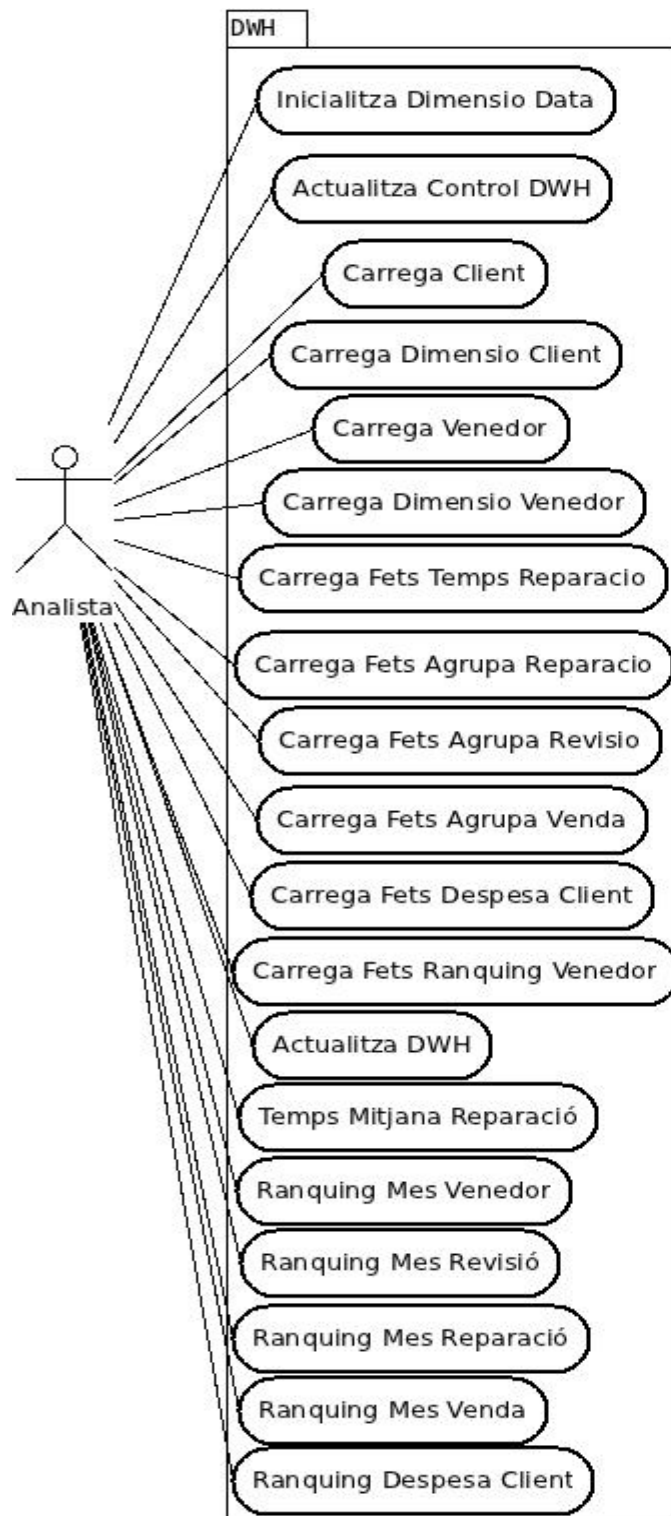
A partir de l'anàlisi de requisits s'ha d'obtenir una estructura de la informació que ajudarà a definir la model de BD independentment del SGBD utilitzat.

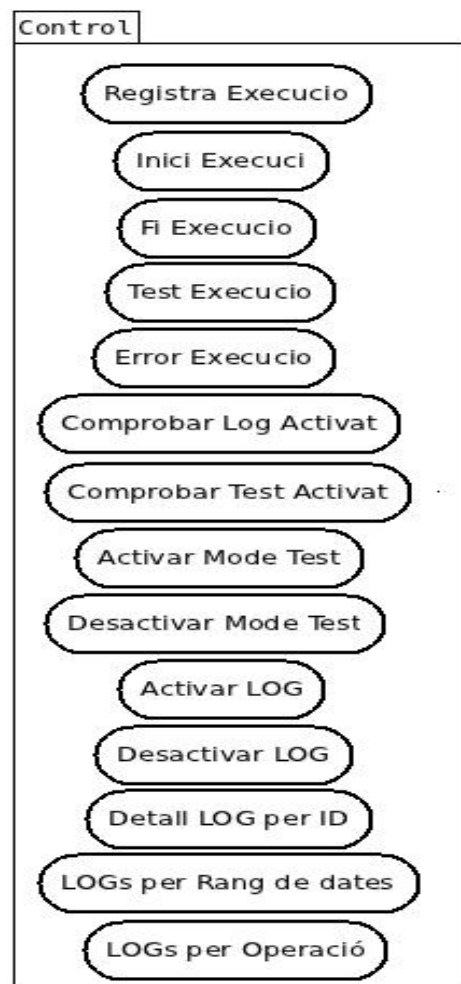
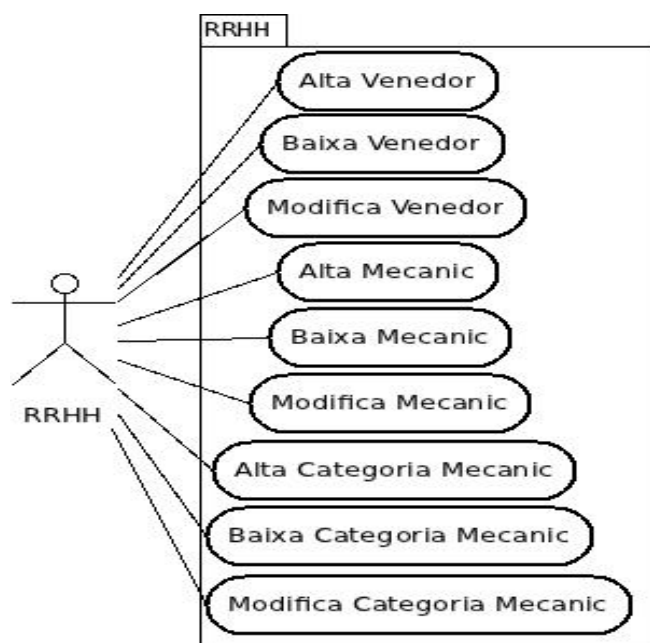
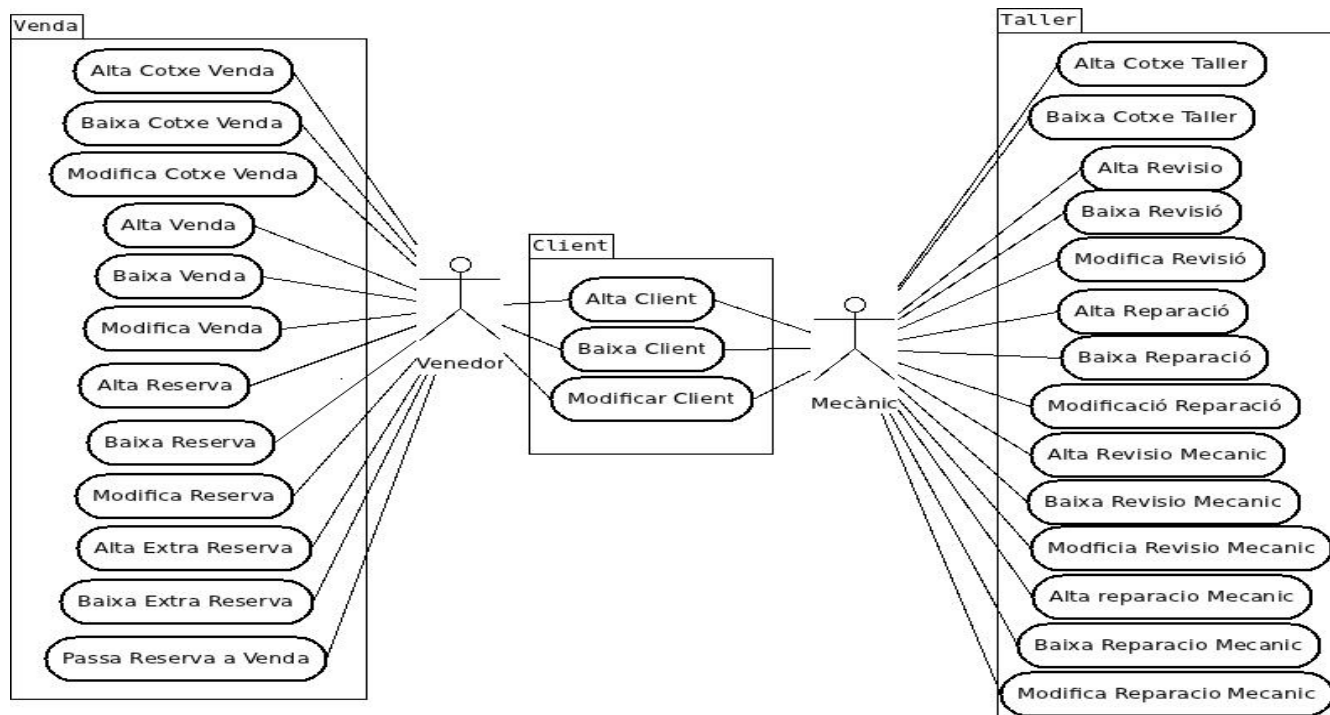
Primer de tot es decideix definir els diferents casos d'ús que són necessaris per a complir els requeriments del client. Després s'utilitza una representació gràfica del disseny mitjançant un diagrama *Entitat-Relació (E/R)* en UML i, finalment, s'analitzarà per separat els diferents elements que hi apareixen: entitats, atributs i relacions.

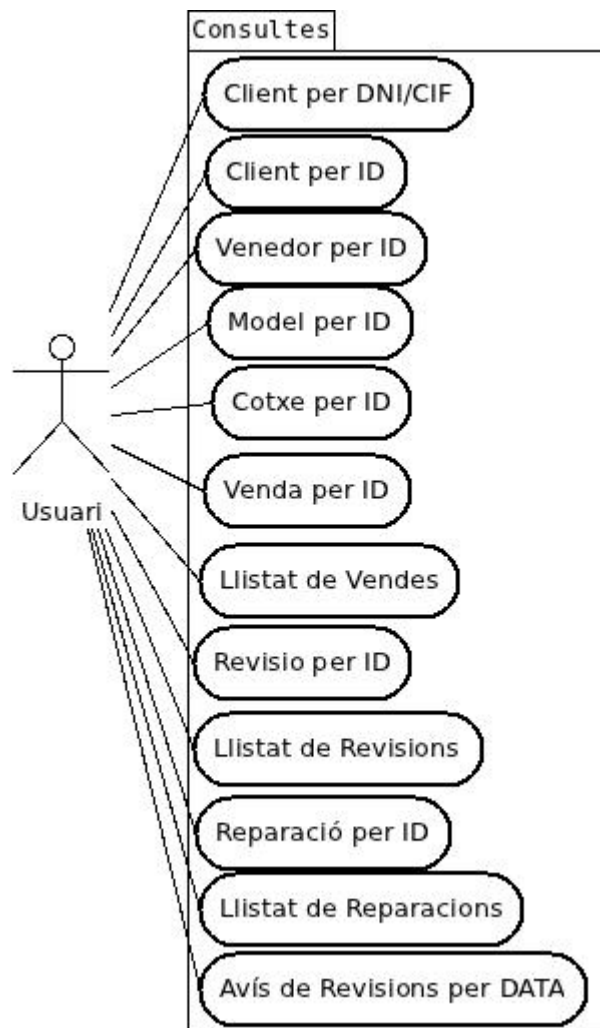
3.1.1.- Casos d'ús

Els actors que apareixen en els casos d'ús són els següents:

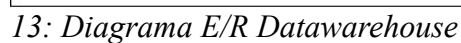
- RRHH: qualcú de recursos humans que pot donar d'alta i de baixa a empleats del taller.
- Administratiu: qualque administratiu que pot encarregar-se de donar d'alta models, marques i extres dels vehicles.
- Venedor: els encarregats de donar d'alta un nou client, de vendre els vehicles i de donar-los d'alta al sistema.
- Mecànic: els encarregats de donar d'alta un nou client, una nova reparació i una nova revisió.
- Usuari: qualsevol usuari que vulgui realitzar consultes.



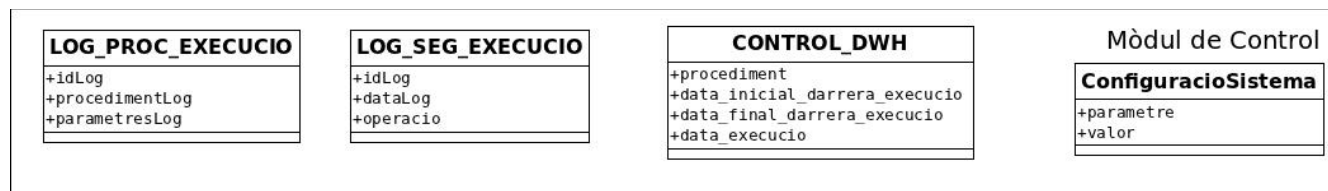




3.1.2.1.-Enunciat



3.1.2.3.-Mòdul Control



3.1.3.- Entitats i atributs

3.1.3.1.- Enunciat

Client: idClient, nomClient, llinatge1, llinatge2, dataNaixementClient, sexeClient, dniCifClient, telefonClient, dataAlta

Venedor: idVenedor, nomVenedor, llinatge1Venedor, llinatge2Venedor, telefonVenedor, dataAlta

Cotxe: idCotxe, bastidorCotxe, matriculaCotxe

Marca: nomMarca

Model: idModel, preuModel, potenciaModel, numPortesModel, cilindradaModel, combustibleModel

Extra: idExtra, nomExtra

CotxesVenda: dataEntrada

Nou:

SegonaMa: dataPrimeraMatriculació, preuSegonaMa

CotxesTaller:

Venda: idVenda, preuVenda, dataReserva

Reserva: idReserva, preuReserva, dataReserva

Revisió: idRevisio, dataRevisio, motiu, esPressupost

Reparació: idReparacio, dataIniciReparacio, dataFiReparacio, motiu, costPecesReparacio, esPressupostReparacio

Mecanic: idMecanic, nomMecanic, llinatge1Mecanic, llinatge2Mecanic

CategoriaMecanic: idCategoriaMecanic, nomCategoriaMecanic, preuCategoriaMecanic

3.1.3.2.- Mòdul Datawarehouse

3.1.3.2.1.- Taules de Dimensions

dData: idData, dataAny, dataMes, dataDia

dClient: idClient, sexeClient

dVenedor: idVenedor

3.1.3.2.2.- Taules de Fets

TempsReparacioCotxe: tempsMitjanaReparacio

RanquingVenedor: vendaAcumulada

MesosRevisio: numRevisio

MesosReparacio: numReparacio

MesosVenda: numVendes

DespesaClient: despesaClient

3.1.3.3.- Mòdul de control

LOG_PROC_EXECUCIO: idLog, procedimentLog, parametresLog

LOG_SEG_EXECUCIO: idLog, dataLog, operacions

CONTROL_DWH: procediment, data_inicial_darrera_execucio, data_final_darrera_execucio, data_execucio

Configuracio_sistema*: parametre, valor

*S'utilitzarà la taula *Configuracio_sistema* per poder afegir paràmetres que permetin ser consultats pels procediments i així, per exemple, poder configurar la BD en mode proves, per exemple.

3.1.4.- Relacions

Les relacions que apareixen en el model *Entitat-Relació* són:

Relació	Descripció
modelReservat (1:N)	En una reserva, es reserva un model concret.
afegit (N:M)	En una reserva, al model s'afegeixen una sèrie de extres.
reservat (1:N)	Una reserva la realitza un client.
reservatV (1:N)	D'una reserva s'encarrega un venedor.
pertany (1:N)	Un model pertany a una marca de vehicles.
esModel (1:N)	Un cotxe és un model concret.
opcio (N:M)	Un model inclou uns extres com a opcionals. Té l'atribut <i>preu</i> .
addicional (N:M)	Un cotxe per vendre pot incloure una sèrie d'extres addicionals.
serie (N:M)	Un model inclou uns extres de sèrie. Té l'atribut <i>preu</i> .
comprat (1:N)	A una venda sempre hi ha un sol client.
cotxeComprat (1:N)	Una venda sempre inclou un vehicle. Un cotxe pot ser venut varies vegades.
haVenut (1:N)	Un venedor és qui ha realitzat una o varies vendes.
reparat (1:N)	Un cotxe pot haver estat reparat varies vegades.
revisat (1:N)	Un cotxe pot haver estat revisat varies vegades.
haReparat (N:M)	Varis mecànics poden participar en varies reparacions. Inclou els atributs <i>numHoresReparacio</i> i <i>preuHoresReparacio</i> .
haRevisat (N:M)	Varis mecànics poden participar en varies revisions. Inclou els atributs <i>numHoresReparacio</i> i <i>preuHoresReparacio</i> .
esCategoria (1:N)	Un mecànic té una categoria determinada, amb el corresponent preu per hora de treball.

3.2.- Disseny lògic

El disseny lògic pren com a punt de partida el model *E/R* per a transformar-lo en un conjunt de relacions i claus a partir d'una sèrie de regles que s'apliquen sobre les entitats i les relacions.

3.2.1.- Model relacional

3.2.1.1.- Enunciat

El model relacional final ha quedat de la següent forma:

Client (idClient, nomClient, llinatge1Client, llinatge2Client, anyNaixementClient, dniCifClient, telefonClient)

Venedor: (idVenedor, nomVenedor, llinatge1Venedor, llinatge2Venedor, telefonVenedor)

Cotxe: (idCotxe, bastidorCotxe, matriculaCotxe)

Model: (idModel, nomMarca, preuModel, potenciaModel, numPortesModel, cilindradaModel, combustibleModel)

CotxesVenda: (idCotxe, dataEntradaCotxe, nouSegonaMa, preuSegonaMa, dataPrimeraMatriculacio)

On {idCotxe} es clau forana de Client(idCotxe)

CotxesTaller: (idCotxe)

On {idCotxe} es clau forana de Client(idCotxe)

Extra: (idExtra, nomExtra, preuExtra)

ExtraModelSerie: (idExtra, idModel)

On {idExtra} es clau forana de Extra(idExtra)

On {idModel} es clau forana de Model(idModel)

ExtraModelOpcio: (idExtra, idModel)

On {idExtra} es clau forana de Extra(idExtra)

On {idModel} es clau forana de Model(idModel)

Venda: (idVenda, idCotxe, idClient, idVenedor, preuVenda, dataVenda)

On {idCotxe} es clau forana de Cotxe(idCotxe)

On {idClient} es clau forana de Client(idClient)

On {idVenedor} es clau forana de Venedor(idVenedor)

Adicional: (idExtra, idCotxe)

On {idExtra} es clau forana de Extra(idExtra)

On {idCotxe} es clau forana de Cotxe(idCotxe)

Reserva: (idReserva, idModel, idClient, idVenedor, preuReserva, dataReserva)

On {idModel} es clau forana de Model(idModel)

On {idClient} es clau forana de Client(idClient)

On {idVenedor} es clau forana de Venedor(idVenedor)

ExtraAfegitReserva: (idReserva, idExtra)

On {idReserva} es clau forana de Reserva(idReserva)

On {idExtra} es clau forana de Extra(idExtra)

Revisio: (idRevisio, idCotxe, dataRevisio)

On {idCotxe} es clau forana de Cotxe(idCotxe)

Reparacio: (idRevisio, idCotxe, dataIniciReparacio, dataFiReparacio, costPecesReparacio, esPressupostReparacio)

On {idCotxe} es clau forana de Cotxe(idCotxe)

haReparat: (idReparacio, idMecanic, numHoresReparacio, preuHoresReparacio)

On {idReparacio} es clau forana de Reparacio(idReparacio)

On {idMecanic} es clau forana de Mecanic(idMecanic)

haRevisat: (idRevisio, idMecanic, numHoresRevisio, preuHoresRevisio)

On {idRevisio} es clau forana de Reparacio(idRevisio)

On {idMecanic} es clau forana de Mecanic(idMecanic)

Mecanic: (idMecanic, nomMecanic, llinatge1Mecanic, llinatge2Mecanic, idCategoriaMecanic)

On {idCategoriaMecanic} es clau forana de CategoriaMecanic(idCategoriaMecanic)

CategoriaMecanic: (idCategoriaMecanic, nomCategoriaMecanic, preuHoraCategoriaMecanic)

3.2.1.2.- Mòdul Datawarehouse

3.2.1.2.1.- Taules de Dimensions

dData (idData, dataAny, dataMes, dataDia)

dClient (idClient, sexeClient)

dVenedor: (idVenedor)

3.2.1.2.2.- Taules de Fets

TempsReparacioCotxe (idData, tempsMitjanaReparacio)

On {idData} es clau forana de dData(idData)

RanquingVenedor (idData, idVenedor, venda)

On {idData} es clau forana de dData(idData)

On {idVenedor} es clau forana de dVenedor(idVenedor)

MesosRevisio (idData, numRevisio)

On {idData} es clau forana de dData(idData)

MesosReparacio (idData, numReparacio)

On {idData} es clau forana de dData(idData)

MesosVenda (idData, numVendes)

On {idData} es clau forana de dData(idData)

DespesaClient (idData, idClient, despesaClient)

On {idData} es clau forana de dData(idData)

On {idClient} es clau forana de dClient(idClient)

3.3.- Disseny físic

En aquesta etapa es transforma l'estructura obtinguda en el disseny lògic amb l'objectiu d'adaptar-lo als aspectes d'implementació físics propis del SGBD que s'utilitza. En aquest cas ORACLE.

3.3.1.- Creació de la BD

Existeixen varies formes per crear una BD amb Oracle:

- Utilitzar la comanda CREATE DATABASE des d'un *sqlplus*.
- Utilitzar el Database Configuration Assistant (DBCA).

Com que la versió d'Oracle és una *XE* que ve amb un paquet *rpm* i després d'adaptar-lo als paquets que utilitza *ubuntu* (deb) no ha acabat de funcionar del tot bé, s'ha hagut de crear la BD manualment executant el següent script (el qual es pot trobar al fitxer

3_3_1_DissenyFisic_CreacioBD.sql):

```
CREATE DATABASE XE
  USER SYS IDENTIFIED BY uocpass
  USER SYSTEM IDENTIFIED BY uocpass
  LOGFILE GROUP 1 ('/u01/app/oracle/oradata/XE/redo01.log') SIZE 100M,
          GROUP 2 ('/u01/app/oracle/oradata/XE/redo02.log') SIZE 100M,
          GROUP 3 ('/u01/app/oracle/oradata/XE/redo03.log') SIZE 100M
  MAXLOGFILES 5
  MAXLOGMEMBERS 5
  MAXLOGHISTORY 1
  MAXDATAFILES 100
  CHARACTER SET US7ASCII
  NATIONAL CHARACTER SET AL16UTF16
  EXTENT MANAGEMENT LOCAL
  DATAFILE '/u01/app/oracle/oradata/XE/system01.dbf' SIZE 325M REUSE
  SYSAUX DATAFILE '/u01/app/oracle/oradata/XE/sysaux01.dbf' SIZE 325M REUSE
  DEFAULT TABLESPACE users
    DATAFILE '/u01/app/oracle/oradata/XE/users01.dbf'
    SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED
  DEFAULT TEMPORARY TABLESPACE temp
    TEMPFILE '/u01/app/oracle/oradata/XE/temp01.dbf'
    SIZE 20M REUSE
  UNDO TABLESPACE UNDOTBS1
    DATAFILE '/u01/app/oracle/oradata/XE/undotbs01_01.dbf'
    SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
@?/rdbms/admin/catalog.sql
@?/rdbms/admin/catproc.sql
@?/sqlplus/admin/pupbld.sql
```

3.3.2.- Creació del tablespace

El *tablespace* és la unitat d'emmagatzemament lògica que utilitza Oracle per desar els objectes de la BD. El *tablespace* està compost per un conjunt d'un o més fitxers físics que es creen al disc dur anomenats *datafiles*.

La sentència que es segueix per crear el *tablespace* és la següent (es pot trobar al fitxer *3_3_6_DissenyFisic_CreacioTablespace.sql*):

```
CREATE TABLESPACE TFC11_12
  DATAFILE '/u01/app/oracle/oradata/XE/tfc11_12.dbf'
  SIZE 250M
  AUTOEXTEND OFF;
```

3.3.3.- Creació d'usuari

El sistema necessita un usuari de connexió que permeti accedir a la BD. Aquest usuari tindrà els rols *CONNECT* i *RESOURCE*, els quals donen el privilegis necessaris per poder implementar i executar el sistema correctament.

La sentència de creació de l'usuari és la següent (es pot trobar al fitxer

3_3_3_DissenyFisic_CreacioUsuari.sql):

```
CREATE USER TFCUSER  
IDENTIFIED BY TFCUSER  
DEFAULT TABLESPACE TFC11_12  
TEMPORARY TABLESPACE TEMP;  
  
GRAN CONNECT, RESOURCE TO TFCUSER;
```

3.3.4.- Creació de taules

Dins del fitxer *3_3_4_DissenyFisic_CreacioTaules.sql* es troben els diferents scripts de creació de les taules dels diferents apartats.

3.3.5.- Creació d'índexs

Quan es defineixen claus primàries a les taules, Oracle ja crea un índex nul i únic. De forma addicional es crearan índexs a mesura que es detectin com a necessaris durant la implementació.

3.3.6.- Creació de seqüències

Es decideix utilitzar seqüències per poder obtenir fàcilment identificadors únics en els casos que siguin necessaris. Dins del fitxer *3_3_6_DissenyFisic_Sequencies.sql* es poden trobar les sentències per crear les seqüències.

4.- Implementació

S'han implementat set paquets diferents (amb els seus corresponents procediments) per poder complir les funcionalitats sol·licitades per part del client. Per a crear tots els paquets a l'hora es pot executar el fitxer *4_Implementacio.sql*.

Hi ha dues característiques importants a tenir en compte sobre el funcionament del sistema implementat:

1. Totes les excepcions són enviades a una taula de *LOG*. Aquest LOG es pot activar o desactivar en qualsevol moment. Si hi ha qualche error, serà suficient activar el LOG per detectar on es troba l'arrel del problema.
2. El sistema té l'opció de posar-se en mode test. Si es troba en mode test, no es realitzarà cap modificació (ni inserció, ni actualització ni esborrament de dades) sobre el mateix.

4.1.-Package Control

Aquest paquet proporciona els diferents procediments necessaris que permeten habilitar i deshabilitar tant el LOG com el mode test, així com inserir en el LOG els diferents estats.

El codi per crear aquest paquet es pot trobar al fitxer *4_1_Implementacio_PkgControl.sql*.

7.1.1.-REGISTRA_EXECUCIO

Propòsit
- Aquesta funció s'encarrega d'incrementar la seqüència de LOG i inserir una nova entrada dins de la taula <i>LOG_PROC_EXECUCIO</i> .
Paràmetres
- <i>procediment VARCHAR2</i> : nom del procediment on es crida la funció. - <i>parametres VARCHAR2</i> : paràmetres amb els quals s'ha executat el procediment on es crida la funció.
Retorna
- Un numèric corresponent al identificador de LOG assignat per la seqüència.

4.1.2.-INICI_EXECUCIO

Propòsit
- Aquest procediment rep com a paràmetre un identificador de log i insereix el text <i>INICI</i> dins del camp <i>operació</i> de la taula <i>LOG_SEG_EXECUCIO</i> .
Paràmetres
- <i>idLog NUMERIC</i> : identificador únic de log

4.1.3.-FI_EXECUCIO

Propòsit
- Aquest procediment rep com a paràmetre un identificador de log i insereix el text FI dins del camp <i>operació</i> de la taula <i>LOG_SEG_EXECUCIO</i> .
Paràmetres
- <i>idLog NUMERIC</i> : identificador únic de log

4.1.4.-TEST_EXECUCIO

Propòsit
- Aquest procediment s'executa quan el sistema es troba en mode test. Rep com a paràmetre un identificador de LOG i insereix el text TEST dins del camp <i>operació</i> de la taula <i>LOG_SEG_EXECUCIO</i> .
Paràmetres
- <i>idLog NUMERIC</i> : identificador únic de log

4.1.5.-ERROR_EXECUCIO

Propòsit
- Aquest procediment s'executa quan es produeix un error. Rep com a paràmetre un identificador de log i insereix la paraula ERROR seguida del missatge d'error que ha retornat la BD dins del camp <i>operació</i> de la taula <i>LOG_SEG_EXECUCIO</i>
Paràmetres
- <i>idLog NUMERIC</i> : identificador únic de log

4.1.6.-LOG_ACTIVAT

Propòsit
- Aquesta funció torna <i>true</i> si el paràmetre de la taula de <i>configuracio_sistema</i> indica que el <i>LOG</i> està activat i <i>false</i> en cas contrari.
Paràmetres
- No en té.
Retorna
- Un valor booleà.

4.1.7.-MODE_TEST

Propòsit
- Aquesta funció torna <i>true</i> si el paràmetre de la taula de <i>configuracio_sistema</i> indica que el mode test està activat i <i>false</i> en cas contrari.
Paràmetres
- No en té.
Retorna
- Un valor booleà.

4.1.8.-ACTIVA_LOG

Propòsit
- Aquest procediment modifica el paràmetre de la taula <i>configuracio_sistema</i> de manera que el <i>LOG</i> quedi activat.
Paràmetres
- No en té.

4.1.9.-DESACTIVA_LOG

Propòsit
- Aquest procediment modifica el paràmetre de la taula <i>configuracio_sistema</i> de manera que el <i>LOG</i> quedi desactivat.
Paràmetres
- No en té.

4.1.10.-ACTIVA_TEST

Propòsit
- Aquest procediment modifica el paràmetre de la taula <i>configuracio_sistema</i> de manera que el mode test quedi activat i el sistema deixi d'enregistrar canvis.
Paràmetres
- No en té.

4.1.11.-DESACTIVA_TEST

Propòsit
- Aquest procediment modifica el paràmetre de la taula <i>configuracio_sistema</i> de manera que el mode test quedi desactivat i el sistema comenci a enregistrar els canvis.
Paràmetres
-

4.1.12.-LOG_ID

Propòsit
- Aquest procediment retorna el detall de les operacions que hi ha enregistrades per a un idLog concret.
Paràmetres
- <i>PidLog NUMERIC</i> : identificador de log que es vol consultar.

4.1.13.-LOG_RANG_DATA

Propòsit
- Aquest procediment retorna el detall de les operacions que hi ha enregistrades dins d'un rang de temps concret que s'ha passat per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial. - <i>PdataFi DATE</i> : data final.

4.1.14.-LOG_CERCA_OPERACIO

Propòsit
- Aquest procediment retorna el detall de les operacions que hi ha enregistrades i que contenen un text concret passat per paràmetre.
Paràmetres
- <i>Ptext Varchar2</i> : text que es cercarà en el camp operació dels logs.

4.2.-Package Client

El codi per crear aquest paquet es pot trobar al fitxer *4_2_Implementacio_PkgClient.sql*.

4.2.1.-ALTA_CLIENT

Propòsit
- Aquest procediment dóna d'alta un nou client en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència.
Paràmetres
- <i>PidClient NUMERIC</i> : identificador únic del client. - <i>PnomClient VARCHAR2</i> : nom del client. - <i>Pllinatge1Client VARCHAR2</i> : primer llinatge del client. - <i>Pllinatge2Client VARCHAR2</i> : segon llinatge del client. - <i>PdataNaixementClient DATE</i> : data de naixement del client. - <i>PsexeClient VARCHAR2</i> : sexe del client. - <i>PdniCifClient VARCHAR2</i> : DNI del client. - <i>PtelefonClient NUMERIC</i> : nombre de telèfon de contacte del client. - <i>PdataAlta DATE</i> : data d'alta del client.

4.2.2.-BAIXA_CLIENT

Propòsit
- Aquest procediment elimina un client amb un identificador concret del sistema.
Paràmetres
- <i>PidClient NUMERIC</i> : identificador únic del client que s'ha d'esborrar.

4.2.3.-MODIFICA_CLIENT

Propòsit
- Aquest procediment modifica el contingut del client amb un <i>idClient</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidClient NUMERIC</i> : identificador únic del client. - <i>PnomClient VARCHAR2</i> : nom del client. - <i>Pllinatge1Client VARCHAR2</i> : primer llinatge del client. - <i>Pllinatge2Client VARCHAR2</i> : segon llinatge del client. - <i>PdataNaixementClient DATE</i> : data de naixement del client. - <i>PsexeClient VARCHAR2</i> : sexe del client. - <i>PdniCifClient VARCHAR2</i> : DNI del client. - <i>PtelefonClient NUMERIC</i> : nombre de telèfon de contacte del client. - <i>PdataAlta DATE</i> : data d'alta del client.

4.3.-Package Cotxes

El codi per crear aquest paquet es pot trobar al fitxer *4_3_Implementacio_PkgCotxes.sql*.

4.3.1.-ALTA_MARCA

Propòsit
- Aquest procediment dóna d'alta una nova marca de cotxes en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidMarca NUMERIC</i> : identificador únic de la marca. - <i>PnomMarca VARCHAR2</i> : nom de la marca.

4.3.2.-BAIXA_MARCA

Propòsit
- Aquest procediment esborra una marca amb un identificador concret del sistema.
Paràmetres
- <i>PidMarca NUMERIC</i> : identificador únic de la marca.

4.3.3.-MODIFICA_MARCA

Propòsit
- Aquest procediment modifica el contingut de la marca amb un <i>idMarca</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidMarca NUMERIC</i> : identificador únic de la marca. - <i>PnomMarca VARCHAR2</i> : nom de la marca.

4.3.4.-ALTA_MODEL

Propòsit
- Aquest procediment dona d'alta un nou model en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidModel NUMERIC</i> : identificador únic del model. - <i>PnomModel VARCHAR2</i> : nom del model. - <i>PpreuModel NUMBER</i> : preu del model base. - <i>PpotenciaModel NUMBER</i> : potència (en cavalls) del model. - <i>PnumPortesModel NUMBER</i> : nombre de portes del model. - <i>PcombustibleModel VARCHAR2</i> : tipus de combustible utilitzat pel model. - <i>PidMarca NUMERIC</i> : identificador únic de la marca a la qual pertany el model.

4.3.5.-BAIXA_MODEL

Propòsit
- Aquest procediment esborra un model amb un identificador concret del sistema.
Paràmetres
- <i>PidModel NUMERIC</i> : identificador únic del model que es vol esborrar.

4.3.6.-MODIFICA_MODEL

Propòsit
- Aquest procediment modifica el contingut del model amb un <i>idModel</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidModel NUMERIC</i> : identificador únic del model. - <i>PnomModel VARCHAR2</i> : nom del model. - <i>PpreuModel NUMBER</i> : preu del model base. - <i>PpotenciaModel NUMBER</i> : potència (en cavalls) del model. - <i>PnumPortesModel NUMBER</i> : nombre de portes del model. - <i>PcombustibleModel VARCHAR2</i> : tipus de combustible utilitzat pel model. - <i>PidMarca NUMERIC</i> : identificador únic de la marca a la qual pertany el model.

4.3.7.-ALTA_EXTRA

Propòsit
- Aquest procediment dona d'alta un nou extra en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència.
Paràmetres
- <i>PidExtra NUMERIC</i> : identificador únic del extra. - <i>PnomExtra VARCHAR2</i> : nom del extra.

4.3.8.-BAIXA_EXTRA

Propòsit
- Aquest procediment esborra una estra de la taula amb un identificador concret.
Paràmetres
- <i>PidExtra NUMERIC</i> : identificador únic del extra.

4.3.9.-MODIFICA_EXTRA

Propòsit
- Aquest procediment modifica el contingut del extra amb un <i>idExtra</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidExtra NUMERIC</i> : identificador únic del extra. - <i>PnomExtra VARCHAR2</i> : nom del extra.

4.3.10.-AFEGIR_EXTRA_MODEL

Propòsit
- Aquest procediment relaciona un extra amb un model concret en el sistema.
Paràmetres
- <i>PidExtra NUMERIC</i> : identificador únic del extra. - <i>PidModel NUMERIC</i> : identificador únic del model de cotxe. - <i>PpreuExtra NUMBER</i> : preu del extra en el model de cotxe. - <i>PserieOpcio VARCHAR2</i> : indica qui és una opció en el model o si ho inclou de sèrie.

4.3.11.-ELIMINAR_EXTRA_MODEL

Propòsit
- Aquest procediment esborra de la taula una entrada amb un identificador de extra <i>idExtra</i> i un identificador <i>idModel</i> concrets.
Paràmetres
- <i>PidExtra NUMERIC</i> : identificador únic del extra. - <i>PidModel NUMERIC</i> : identificador únic del model de cotxe.

4.3.12.-MODIFICAR_EXTRA_MODEL

Propòsit
- Aquest procediment modifica el contingut de la taula amb un identificador de extra <i>idExtra</i> i un identificador de model <i>idModel</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidExtra NUMERIC</i> : identificador únic del extra. - <i>PidModel NUMERIC</i> : identificador únic del model de cotxe. - <i>PpreuExtra NUMBER</i> : preu del extra en el model de cotxe. - <i>PserieOpcio VARCHAR2</i> : indica qui és una opció en el model o si ho inclou de sèrie.

4.3.13.-ALTA_COTXE

Propòsit
- Aquest procediment dona d'alta un nou cotxe amb un identificador de cotxe <i>idCotxe</i> en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PbastidorCotxe VARCHAR2</i> : número de bastidor del cotxe. - <i>PmatriculaCotxe VARCHAR2</i> : número de la matricula del cotxe. - <i>PidModel NUMERIC</i> : identificador únic del model de cotxe.

4.3.14.-BAIXA_COTXE

Propòsit
- Aquest procediment dona de baixa en el sistema un cotxe amb un <i>idCotxe</i> concret.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe.

4.3.15.-MODIFICA_COTXE

Propòsit
- Aquest procediment modifica el contingut del cotxe amb un <i>idCotxe</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PbastidorCotxe VARCHAR2</i> : número de bastidor del cotxe. - <i>PmatriculaCotxe VARCHAR2</i> : número de la matricula del cotxe. - <i>PidModel NUMERIC</i> : identificador únic del model de cotxe.

4.4.-Package DWH

El codi per crear aquest paquet es pot trobar al fitxer *4_4_Implementacio_PkgDWH.sql*.

4.4.1.-INICIALITZA_DIMENSIO_DATA

Propòsit
- Aquest procediment s'encarrega d'inicialitzar la dimensió data dins del DWH. Carrega dins la dimensió les dates corresponents al període que compren entre <i>PdataInici</i> i <i>PdataFi</i> .
Paràmetres
- <i>PdataInici DATE</i> : data inicial en la s'ha d'inicialitzar la dimensió data. - <i>PdataInici DATE</i> : data final en la s'ha d'inicialitzar la dimensió data.

4.4.2.-ACTUALITZA_CONTROL_DWH

Propòsit
- Aquest procediment s'encarrega d'actualitzar les dades corresponents al procediment <i>Pprocediment</i> dins de la taula <i>CONTROL_DWH</i> . S'utilitza per saber quin rang de dades es va executar la darrera vegada.
Paràmetres
- <i>Pprocediment VARCHAR2</i> : nom del procediment des d'on s'executa. - <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment <i>Pprocediment</i> . - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment <i>Pprocediment</i> .

4.4.3.-CARREGA_CLIENT

Propòsit
- Aquest procediment insereix dins de la dimensió client un client amb el sexe corresponent ja transformat a valor numèric.
Paràmetres
- <i>idClient NUMERIC</i> : identificador únic del client a inserir. - <i>sexeClient NUMBER</i> : sexe del client transformat a valor numèric.

4.4.4.-CARREGA_DIMENSIO_CLIENT

Propòsit
- Aquest procediment selecciona els clients donats d'alta entre les dates passades per paràmetre, transforma les dades necessàries i les insereix dins de la dimensió client.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.5.-CARREGA_VENEDOR

Propòsit
- Aquest procediment insereix dins de la dimensió venedor un venedor.
Paràmetres
- <i>idVenedor NUMERIC</i> : identificador únic del venedor a inserir.

4.4.6.-CARREGA_DIMENSIO_VENEDOR

Propòsit
- Aquest procediment selecciona els venedors donats d'alta entre les dates passades per paràmetre, transforma les dades necessàries i les insereix dins de la dimensió venedor.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.7.-CARREGA_FETS_TEMPSPREPARACIO

Propòsit
- Aquest procediment carrega dins de la taula de fets corresponent el temps mitjà de reparacions realitzades per dia entre les dates passades per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.8.-CARREGA_FETS_AGRUPAREPARACIO

Propòsit
- Aquest procediment carrega dins de la taula de fets corresponent el nombre de reparacions realitzades per dia entre les dates passades per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.9.-CARREGA_FETS_AGRUPAREVISIO

Propòsit
- Aquest procediment carrega dins de la taula de fets corresponent el nombre de revisions realitzades per dia entre les dates passades per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.10.-CARREGA_FETS_AGRUPAVENDA

Propòsit
- Aquest procediment carrega dins de la taula de fets corresponent el nombre de vendes realitzades per dia en el període comprés entre les dates passades per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.11.-CARREGA_FETS_DESPESACLIENT

Propòsit
- Aquest procediment carrega dins de la taula de fets corresponent la suma de les despeses per dia realitzades per un client entre les dates passades per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.12.-CARREGA_FETS_RANQUINGVENEDOR

Propòsit
- Aquest procediment carrega dins de la taula de fets corresponent la suma de les vendes realitzades per dia par cada venedor entre les dates passades per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.13.-ACTUALITZA_DWH

Propòsit
- Aquest procediment crida a tots els procediments que actualitzen tant dimensions com fets amb el mateix rang de dates (exceptuant el d'inicialització de la dimensió dates).
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual s'executarà el procediment. - <i>PdataInici DATE</i> : data final amb la qual s'executarà el procediment.

4.4.14.-TEMPSMITJANAREPARACIO

Propòsit
- Aquest procediment mostra per pantalla el temps mitjà de reparació per vehicle que hi ha al taller.
Paràmetres
- No en té.

4.4.15.-RANQUINGMESVENEDOR

Propòsit
- Aquest procediment, per a un mes d'un any concret, mostra per pantalla el rànquing de venedors d'aquell mes ordenat per millor vendes.
Paràmetres
- <i>PANY NUMERIC</i> : número d'any. - <i>PMES NUMERIC</i> : número de mes

4.4.16.-RANQUINGMESREVISIO

Propòsit
- Aquest procediment, per un rang de mesos concrets, mostra per pantalla el nombre de revisions realitzades en aquells mesos.
Paràmetres
- <i>PdataInici NUMERIC</i> : mes inicial (format YYYYMM) amb la qual s'executarà el procediment. - <i>PdataInici NUMERIC</i> : mes final (format YYYYMM) amb la qual s'executarà el procediment.

4.4.17.-RANQUINGMESREPARACIO

Propòsit
- Aquest procediment, per un rang de mesos concrets, mostra per pantalla el nombre de reparacions realitzades en aquells mesos.
Paràmetres
- <i>PdataInici NUMERIC</i> : mes inicial (format YYYYMM) amb la qual s'executarà el procediment. - <i>PdataInici NUMERIC</i> : mes final (format YYYYMM) amb la qual s'executarà el procediment.

4.4.18.-RANQUINGMESVENDA

Propòsit
- Aquest procediment, per un rang de mesos concrets, mostra per pantalla el nombre de vendes realitzades en aquells mesos.
Paràmetres
- <i>PdataInici NUMERIC</i> : mes inicial (format YYYYMM) amb la qual s'executarà el procediment. - <i>PdataInici NUMERIC</i> : mes final (format YYYYMM) amb la qual s'executarà el procediment.

4.4.19.-RANQUINGDESPESACLIENT

Propòsit
- Aquest procediment mostra per pantalla el llistat global de clients que han realitzat despeses al llarg del temps, la quantitat que han pagat i la posició que ocupen, ordenat de major a menor despesa.
Paràmetres
- No en té

4.5.-Package RRHH

El codi per crear aquest paquet es pot trobar al fitxer *4_5_Implementacio_PkgRRHH.sql*.

4.5.1.-ALTA_VENEDOR

Propòsit
- Aquest procediment dóna d'alta un nou venedor en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidVenedor NUMERIC</i> : identificador únic del venedor. - <i>PnomVenedor VARCHAR2</i> : nom del venedor. - <i>Pllinatge1Venedor VARCHAR2</i> : primer llinatge del venedor. - <i>Pllinatge2Venedor VARCHAR2</i> : segon llinatge del venedor. - <i>PtelefonVenedor NUMBER</i> : nombre de telèfon del venedor. - <i>PdataAlta DATE</i> : data d'alta del venedor en el sistema.

4.5.2.-BAIXA_VENEDOR

Propòsit
- Aquest procediment elimina un venedor amb un identificador concret del sistema.
Paràmetres
- <i>PidVenedor NUMERIC</i> : identificador únic del venedor que s'ha d'esborrar.

4.5.3.-MODIFICA_VENEDOR

Propòsit
- Aquest procediment modifica el contingut del venedor amb un identificador de venedor <i>idVenedor</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidVenedor NUMERIC</i> : identificador únic del venedor. - <i>PnomVenedor VARCHAR2</i> : nom del venedor. - <i>Pllinatge1Venedor VARCHAR2</i> : primer llinatge del venedor. - <i>Pllinatge2Venedor VARCHAR2</i> : segon llinatge del venedor. - <i>PtelefonVenedor NUMBER</i> : nombre de telèfon del venedor. - <i>PdataAlta DATE</i> : data d'alta del venedor en el sistema.

4.5.4.-ALTA_MECANIC

Propòsit
- Aquest procediment dóna d'alta un nou mecànic en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidMecanic NUMERIC</i> : identificador únic del mecànic. - <i>PnomMecanic VARCHAR2</i> : nom del mecànic. - <i>Pllinatge1Mecanic VARCHAR2</i> : primer llinatge del mecànic. - <i>Pllinatge2Mecanic VARCHAR2</i> : segon llinatge del mecànic. - <i>PidCatMecanic NUMERIC</i> : categoria a la qual pertany el mecànic.

4.5.5.-BAIXA_MECANIC

Propòsit
- Aquest procediment elimina un mecànic amb un identificador concret del sistema.
Paràmetres
- <i>PidMecanic NUMERIC</i> : identificador únic del mecànic que s'ha d'esborrar.

4.5.6.-MODIFICA_MECANIC

Propòsit
- Aquest procediment modifica el contingut del mecànic amb un identificador <i>idMecanic</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidMecanic NUMERIC</i> : identificador únic del mecànic. - <i>PnomMecanic VARCHAR2</i> : nom del mecànic. - <i>Pllinatge1Mecanic VARCHAR2</i> : primer llinatge del mecànic. - <i>Pllinatge2Mecanic VARCHAR2</i> : segon llinatge del mecànic. - <i>PidCatMecanic NUMERIC</i> : categoria a la qual pertany el mecànic.

4.5.7.-ALTA_CATEGORIA_MECANIC

Propòsit
- Aquest procediment dona d'alta una nova categoria de mecànic en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidCategoriaMecanic NUMERIC</i> : identificador únic de la categoria de mecànic. - <i>PnomCategoriaMecanic VARCHAR2</i> : nom de la categoria de mecànic. - <i>PpreuHoraCategoriaMecanic NUMERIC</i> : preu per hora que costa un mecànic d'aquesta categoria.

4.5.8.-BAIXA_CATEGORIA_MECANIC

Propòsit
- Aquest procediment elimina una categoria de mecànic amb un identificador concret del sistema.
Paràmetres
- <i>PidCatMecanic NUMERIC</i> : identificador únic de la categoria de mecànic que s'ha d'esborrar.

4.5.9.-MODIFICA_CATEGORIA_MECANIC

Propòsit
- Aquest procediment modifica el contingut de la categoria de mecànic amb un identificador <i>idCatMecanic</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidCategoriaMecanic NUMERIC</i> : identificador únic de la categoria de mecànic. - <i>PnomCategoriaMecanic VARCHAR2</i> : nom de la categoria de mecànic. - <i>PpreuHoraCategoriaMecanic NUMERIC</i> : preu per hora que costa un mecànic d'aquesta categoria.

4.6.-Package Taller

El codi per crear aquest paquet es pot trobar al fitxer *4_6_Implementacio_PkgTaller.sql*.

4.6.1.-ALTA_COTXE_TALLER

Propòsit
- Aquest procediment relaciona un cotxe que es troba en el taller amb un client concret.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PidClient NUMERIC</i> : identificador únic del client.

4.6.2.-BAIXA_COTXE_TALLER

Propòsit
- Aquest procediment esborra un vehicle del taller amb un identificador únic.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe.

4.6.3.-ALTA_REVISIO

Propòsit
- Aquest procediment dóna d'alta una nova revisió en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidRevisio NUMERIC</i> : identificador únic de la revisió. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PdataRevisio DATE</i> : data en la qual es realitzar la revisió. - <i>PesPressupost VARCHAR2</i> : indica si els costos associats a la revisió són un pressupost o no.

4.6.4.-BAIXA_REVISIO

Propòsit
- Aquest procediment elimina una revisió amb un identificador concret del sistema.
Paràmetres
- <i>PidRevisio NUMERIC</i> : identificador únic de la revisió.

4.6.5.-MODIFICA_REVISIO

Propòsit
- Aquest procediment modifica el contingut d'una revisió amb un identificador <i>PidRevisio</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidRevisio NUMERIC</i> : identificador únic de la revisió. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PdataRevisio DATE</i> : data en la qual es realitzar la revisió. - <i>PesPressupost VARCHAR2</i> : indica si els costos associats a la revisió són un pressupost o no.

4.6.6.-ALTA_REPARACIO

Propòsit
- Aquest procediment dóna d'alta una nova reparació en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidReparació NUMERIC</i> : identificador únic de la reparació. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PdataIniciReparació DATE</i> : data en la qual arriba al taller el vehicle. - <i>PdataFiReparació DATE</i> : data en la qual arriba al taller el vehicle. - <i>PcostPecesReparacio NUMERIC</i> : cost total de les peces que s'han utilitzat en la reparació. - <i>PesPressupost VARCHAR2</i> : indica si els costos associats a la reparació són un pressupost o no.

4.6.7.-BAIXA_REPARACIO

Propòsit
- Aquest procediment elimina una reparació amb un identificador concret del sistema.
Paràmetres
- <i>PidReparació NUMERIC</i> : identificador únic de la reparació.

4.6.8.-MODIFICA_REPARACIO

Propòsit
- Aquest procediment modifica el contingut d'una reparació amb un identificador <i>idReparació</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidReparació NUMERIC</i> : identificador únic de la reparació. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PdataIniciReparació DATE</i> : data en la qual arriba al taller el vehicle. - <i>PdataFiReparació DATE</i> : data en la qual arriba al taller el vehicle. - <i>PcostPecesReparacio NUMERIC</i> : cost total de les peces que s'han utilitzat en la reparació. - <i>PesPressupost VARCHAR2</i> : indica si els costos associats a la reparació són un pressupost o no.

4.6.9.-ALTA_REPARACIO_MECANIC

Propòsit
- Aquest procediment insereix en el sistema el nombre d'hores que ha realitzat un mecànic en una reparació en concret amb el preu per hora de la seva categoria en el moment que s'introdueix.
Paràmetres
- <i>PidReparació NUMERIC</i> : identificador únic de la reparació. - <i>PidMecanic NUMERIC</i> : identificador únic del cotxe. - <i>PnumHoresReparacio NUMERIC</i> : nombre d'hores dedicades.

4.6.10.-BAIXA_REPARACIO_MECANIC

Propòsit
- Aquest procediment elimina les hores realitzades per un mecànic en una reparació.
Paràmetres
- <i>PidReparació NUMERIC</i> : identificador únic de la reparació. - <i>PidMecanic NUMERIC</i> : identificador únic del mecànic.

4.6.11.-MODIFICA_REPARACIO_MECANIC

Propòsit
- Aquest procediment modifica el nombre d'hores i el preu per hora d'un mecànic que ha realitzat en una reparació en concret.
Paràmetres
- <i>PidReparacio NUMERIC</i> : identificador únic de la reparació. - <i>PidMecanic NUMERIC</i> : identificador únic del cotxe. - <i>PnumHoresReparacio NUMERIC</i> : nombre d'hores dedicades. - <i>PpreuHoresReparacio NUMERIC</i> : preu per hora que se li vol assignar al mecànic (per si es vol posar un diferent al qual hi ha a la categoria).

4.6.12.-ALTA_REVISIO_MECANIC

Propòsit
- Aquest procediment insereix en el sistema el nombre d'hores que ha realitzat un mecànic en una revisió en concret amb el preu per hora de la seva categoria en el moment que s'introdueix.
Paràmetres
- <i>PidRevisió NUMERIC</i> : identificador únic de la revisió. - <i>PidMecanic NUMERIC</i> : identificador únic del cotxe. - <i>PnumHoresRevisió NUMERIC</i> : nombre d'hores dedicades.

4.6.13.-BAIXA_REVISIO_MECANIC

Propòsit
- Aquest procediment elimina les hores realitzades per un mecànic en una revisió.
Paràmetres
- <i>PidRevisió NUMERIC</i> : identificador únic de la revisió. - <i>PidMecanic NUMERIC</i> : identificador únic del mecànic.

4.6.14.-MODIFICA_REVISIO_MECANIC

Propòsit
- Aquest procediment modifica el nombre d'hores i el preu per hora d'un mecànic que ha realitzat en una revisió en concret.
Paràmetres
- <i>PidRevisió NUMERIC</i> : identificador únic de la revisió. - <i>PidMecanic NUMERIC</i> : identificador únic del cotxe. - <i>PnumHoresRevisió NUMERIC</i> : nombre d'hores dedicades. - <i>PpreuHoresRevisió NUMERIC</i> : preu per hora que se li vol assignar al mecànic (per si es vol posar un diferent al qual hi ha a la categoria).

4.7.-Package Vendes

El codi per crear aquest paquet es pot trobar al fitxer *4_7_Implementacio_PkgVendes.sql*.

4.7.1.-ALTA_COTXE_VENDA

Propòsit
- Aquest procediment dóna d'alta un cotxe en l'estoc de vendes.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PdataEntradaCotxe DATE</i> : data d'arribada del cotxe per posar-lo en venda. - <i>PnouSegonaMa VARCHAR2</i> : indica si el vehicle és nou i de segona mà. - <i>PpreuSegonaMa NUMBER</i> : si el cotxe es de segona mà, s'indica el preu del mateix. - <i>PdataPrimeraMatriculacio DATE</i> : si el cotxe es troba matriculat, data de la primera matriculació.

4.7.2.-BAIXA_COTXE_VENDA

Propòsit
- Aquest procediment dóna de baixa un cotxe del estoc de vehicles per vendre.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic de la cotxe.

4.7.3.-MODIFICA_COTXE_VENDA

Propòsit
- Aquest procediment actualitza les dades relacionades amb un vehicle amb un identificador <i>idCotxe</i> concret del estoc de vendes.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PdataEntradaCotxe DATE</i> : data d'arribada del cotxe per posar-lo en venda. - <i>PnouSegonaMa VARCHAR2</i> : indica si el vehicle és nou i de segona mà. - <i>PpreuSegonaMa NUMBER</i> : si el cotxe es de segona mà, s'indica el preu del mateix. - <i>PdataPrimeraMatriculacio DATE</i> : si el cotxe es troba matriculat, data de la primera matriculació.

4.7.4.-ALTA_VENDA

Propòsit
- Aquest procediment dóna d'alta una nova venda en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidVenda NUMERIC</i> : identificador únic de la venda. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PidClient NUMERIC</i> : identificador únic del client. - <i>PidVenedor NUMERIC</i> : identificador únic del venedor. - <i>PpreuVenda NUMBER</i> : preu al qual s'ha realitzat la venda. - <i>PdataVenda DATE</i> : data en la qual s'ha realitzat la venda. - <i>PidReserva NUMERIC</i> : identificador únic de la reserva sobre la qual s'ha realitzat la venda.

4.7.5.-BAIXA_VENDA

Propòsit
- Aquest procediment dóna de baixa una venda del sistema.
Paràmetres
- <i>PidVenda NUMERIC</i> : identificador únic de la venda.

4.7.6.-MODIFICA_VENDA

Propòsit
- Aquest procediment modifica el contingut d'una venda amb un identificador <i>idVenda</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidVenda NUMERIC</i> : identificador únic de la venda. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PidClient NUMERIC</i> : identificador únic del client. - <i>PidVenedor NUMERIC</i> : identificador únic del venedor. - <i>PpreuVenda NUMBER</i> : preu al qual s'ha realitzat la venda. - <i>PdataVenda DATE</i> : data en la qual s'ha realitzat la venda. - <i>PidReserva NUMERIC</i> : identificador únic de la reserva sobre la qual s'ha realitzat la venda.

4.7.7.-ALTA_RESERVA

Propòsit
- Aquest procediment dóna d'alta una nova reserva en el sistema i, si no es passa un identificador concret, li associa un identificador de la seqüència corresponent.
Paràmetres
- <i>PidReserva NUMERIC</i> : identificador únic de la reserva. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PidClient NUMERIC</i> : identificador únic del client. - <i>PidVenedor NUMERIC</i> : identificador únic del venedor. - <i>PpreuReserva NUMBER</i> : preu al qual s'ha realitzat la reserva. - <i>PdataReserva DATE</i> : data en la qual s'ha realitzat la reserva. - <i>PdataArribada vehicle DATE</i> : data estimada d'arribada del vehicle reservat.

4.7.8.-BAIXA_RESERVA

Propòsit
- Aquest procediment dóna de baixa una reserva del sistema.
Paràmetres
- <i>PidReserva NUMERIC</i> : identificador únic de la venda.

4.7.9-MODIFICA_RESERVA

Propòsit
- Aquest procediment modifica el contingut d'una reserva amb un identificador <i>idReserva</i> concret. Modifica tots els valors dels camps substituint-los pels que se li passen per paràmetre.
Paràmetres
- <i>PidReserva NUMERIC</i> : identificador únic de la reserva. - <i>PidCotxe NUMERIC</i> : identificador únic del cotxe. - <i>PidClient NUMERIC</i> : identificador únic del client. - <i>PidVenedor NUMERIC</i> : identificador únic del venedor. - <i>PpreuReserva NUMBER</i> : preu al qual s'ha realitzat la reserva. - <i>PdataReserva DATE</i> : data en la qual s'ha realitzat la reserva. - <i>PdataArribada vehicle DATE</i> : data estimada d'arribada del vehicle reservat.

4.7.10.-ALTA_EXTRA_RESERVA

Propòsit
- Aquest procediment relaciona un extra amb una reserva concreta.
Paràmetres
- <i>PidReserva NUMERIC</i> : identificador únic de la reserva. - <i>PidExtra NUMERIC</i> : identificador únic del extra.

4.7.11.-BAIXA_EXTRA_RESERVA

Propòsit
- Aquest procediment relaciona elimina la relació d'una reserva amb un extra.
Paràmetres
- <i>PidReserva NUMERIC</i> : identificador únic de la reserva. - <i>PidExtra NUMERIC</i> : identificador únic del extra.

4.7.12.-PASSA_RESERVA_VENDA

Propòsit
- Aquest procediment s'executa quan una reserva passa a ser una venda. El que fa és donar d'alta el cotxe que ha arribat, associar-hi els extres al cotxe i crear la venda.
Paràmetres
- <i>PidReserva NUMERIC</i> : identificador únic de la reserva. - <i>PbastidorCotxe VARCHAR2</i> : bastidor del cotxe que es ven. - <i>PmatriculaCotxe VARCHAR2</i> : matricula del cotxe, si ja es disposa.

4.8.-Package Consultes

El codi per crear aquest paquet es pot trobar al fitxer *4_8_Implementacio_PkgConsultes.sql*.

4.8.1.-CLIENT_DNICIF

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'un client amb un DNI o CIF concret, depenent de si és una persona física o una empresa.
Paràmetres
- <i>PdniCifClient Varchar2</i> : DNI o CIF del client que es vol consultar.

4.8.2.-CLIENT_ID

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'un client amb un identificador concret.
Paràmetres
- <i>PidClient NUMERIC</i> : identificador del client.

4.8.3.-VENEDOR_ID

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'un venedor amb un identificador concret.
Paràmetres
- <i>PidVenedor NUMERIC</i> : identificador del venedor.

4.8.4.-MODEL_ID

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'un model de cotxe concret, incloent els extrems i opcions que té.
Paràmetres
- <i>PidModel NUMERIC</i> : identificador únic del model a consultar.

4.8.5.-COTXE_ID

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'un cotxe concret, incloent els extres que té.
Paràmetres
- <i>PidCotxe NUMERIC</i> : identificador únic del cotxe a consultar.

4.8.6.-VENDA_ID

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'una venda concreta.
Paràmetres
- <i>PidVenda NUMERIC</i> : identificador únic de la venda a consultar.

4.8.7.-LLISTAT_VENDES

Propòsit
- Aquest procediment mostra per pantalla el detall de totes les vendes enregistrades en el sistema.
Paràmetres
- No en té.

4.8.8.-REVISIO_ID

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'una revisió concreta.
Paràmetres
- <i>PidRevisio NUMERIC</i> : identificador únic de la revisió a consultar.

4.8.9.-LLISTAT_REVISIONS

Propòsit
- Aquest procediment mostra per pantalla el detall de totes les revisions enregistrades en el sistema.
Paràmetres
- No en té.

4.8.8.-REPARACIO_ID

Propòsit
- Aquest procediment mostra per pantalla tota la informació d'una reparació concreta.
Paràmetres
- <i>PidReparacio NUMERIC</i> : identificador únic de la reparació a consultar.

4.8.9.-LLISTAT_REPARACIONS

Propòsit
- Aquest procediment mostra per pantalla el detall de totes les reparacions enregistrades en el sistema.
Paràmetres
- No en té.

4.8.10.-AVISAREVISIONSDATA

Propòsit
- Aquest procediment mostra per pantalla el llistat de revisions que s'han de realitzar dins del període passat per paràmetre.
Paràmetres
- <i>PdataInici DATE</i> : data inicial amb la qual es crida el procediment. - <i>PdataFi DATE</i> : data final amb la qual es crida el procediment.

5.-Proves

Per a provar el correcte funcionament del sistema s'han creat una sèrie de scripts que serveixen per comprovar el correcte funcionament del sistema.

Cada un dels fitxers al executar-se genera un altre fitxer amb el mateix nom però acabat en “.log” que mostra el resultat de l'execució.

5.1.-Càrrega inicial de dades

Amb el fitxer *5_1_Proves_CarregaInicial.sql* es carreguen les dades a dues taules que necessiten ser inicialitzades per a poder utilitzar el sistema:

- CONFIGURACIO SISTEMA: s'insereixen els registres que permetran determinar si el sistema ha de desar el log (LOG_ACTIVAT) i si el sistema es troba en mode test (MODE_TEST).
- CONTROL_DWH: té un registre per a cada procediment que carrega alguna dimensió o fet en el mòdul del Datawarehouse. Serveix per dur un control del que s'ha carregat.

5.2.-Procediments ABM

Amb el fitxer *5_2_Proves_ProcedimentsABM.sql* es prova cada un dels procediments d'alta, baixa i modificació de dades que s'han implementat:

- Categoria Mecànic: es realitzen 4 altes, 1 baixa i 1 modificació.
- Mecànic: es realitzen 7 altes, 1 baixa i 1 modificació.
- Venedor: es realitzen 7 altes, 1 baixa i 1 modificació.
- Client: es realitzen 10 altes, 1 baixa i 1 modificació.
- Marca: es realitzen 6 altes, 1 baixa i 1 modificació.
- Model: es realitzen 16 altes, 1 baixa i 1 modificació.
- Extra: es realitzen 9 altes, 1 baixa i 1 modificació.
- Afegir_extra_model: es realitzen 61 altes, 1 baixa i 1 modificació.
- Cotxe: es realitzen 18 altes, 1 baixa i 1 modificació.
- Cotxes_venda: es realitzen 16 altes i 2 modificacions. Dins de l'execució del procediment *Alta_venda* s'executen les baixes del *cotxes_venda*.
- Reserva: es realitzen 4 altes, 1 baixa i una modificació.
- Extra_reserva: es realitzen 7 altes i 1 baixa.
- Passa_reserva_venda: es passen 3 reserves a vendes.
- Venda: es realitzen 16 altes, 1 baixa i 1 modificació.
- Cotxe_taller: es realitzen 9 altes i 1 baixa.
- Revisio: es realitzen 10 altes, 1 baixa i 1 modificació.
- Reparacio: es realitzen 11 altes, 1 baixa i 1 modificació.
- Hareparat: es realitzen 26 altes, 2 baixes i 1 modificació.
- Harevisat: es realitzen 26 altes, 2 baixes i 1 modificació.

5.3.-Consultes

En el fitxer *5_3_Proves_Consultes.sql* es troben totes les consultes que apareixen en aquest punt. Si s'executa genera un fitxer de log amb els resultats que apareixen a continuació.

5.3.1.-Client per DNI o CIF

Per a consultar un client per DNI o CIF s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.CLIENT_DNICIF('12345681C');
```

El resultat de l'execució és el següent:

```
*****
DADES DEL CLIENT AMB EL DNI 12345681C
*****
ID: 2
NOM: CLIENT2FMOD
LLINATGE 1: CLIENT2L1FMOD
LLINATGE 2:
DATA NAIXAMENT: 06-JUL-90
SEXE: F
DNI/CIF: 12345681C
TELEFON: 666333112
DATA ALTA: 27-FEB-12
```

5.3.2.-Client per ID

Per a consultar un client amb un ID concret s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.CLIENT_ID(1);
```

El resultat de l'execució és el següent:

```
*****
DADES DEL CLIENT AMB EL ID 1
*****
ID: 1
NOM: CLIENT1F
LLINATGE 1: CLIENT1L1F
LLINATGE 2: CLIENT1L2F
DATA NAIXAMENT: 10-JUL-82
SEXE: F
DNI/CIF: 12345680C
TELEFON: 666333111
DATA ALTA: 27-JAN-12
```

5.3.3.-Venedor per ID

Per a consultar un venedor amb un ID concret s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.VENEDOR_ID(2);
```

El resultat de l'execució és el següent:

```
*****
DADES DEL VENEDOR AMB EL ID 2
*****
NOM: VENEDOR2
LLINATGE 1: VENEDOR2L1
LLINATGE 2: VENEDOR2L2
TELEFON: 123456788
DATA ALTA: 10-FEB-12
```

5.3.4.-Model per ID

Per a consultar els detalls d'un model amb un ID concret s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.MODEL_ID(3);
```

El resultat de l'execució és el següent:

```
*****
DADES DEL MODEL AMB EL ID 3
*****
MARCA: PEUGEOT
MODEL: 307 NV
PREU: 16000.5
POTENCIA: 120
NUMPORTES: 5
CILINDRADA: 1.9
COMBUSTIBLE: DIESEL
EXTRA1: AIRBAG      SERIE/OPCIO: S
EXTRA2: SENSOR DARRERA  SERIE/OPCIO: 0
EXTRA3: SENSOR PLUJA  SERIE/OPCIO: S
EXTRA4: ALÇA LLUNES  SERIE/OPCIO: 0
```

5.3.5.-Cotxe per ID

Per a consultar les dades d'un cotxe amb un ID concret s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.COTXE_ID(21);
```

El resultat de l'execució és el següent:

```
*****
DADES DEL COTXE AMB EL ID 21
*****
MARCA: PEUGEOT
MODEL: 307 NV
PREU: 16000.5
POTENCIA: 120
NUMPORTES: 5
CILINDRADA: 2
COMBUSTIBLE: DIESEL
BASTIDORCOTXE: XX211XX
MATRICULACOTXE: 2111WWW
EXTRA1:SENSOR DARRERA  EXTRA2:ALÇA LLUNES
```

5.3.6.-Llistat de vendes

Per a consultar el llistat de totes les vendes que hi ha en el sistema s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.LLISTAT_VENDES;
```

El resultat de l'execució és el següent:

```
*****
LLISTAT DE VENDES
*****
DADES DE LA VENDA AMB EL ID 1
*****
IDVENDA: 1
IDCOTXE: 19
IDCLIENT: 1
IDVENEDOR: 1
PREUVENDA: 15000
DATAVENDA: 11-JAN-13
IDRESERVA: 1
*****
DADES DE LA VENDA AMB EL ID 2
*****
IDVENDA: 2
IDCOTXE: 20
IDCLIENT: 2
IDVENEDOR: 2
PREUVENDA: 20000
DATAVENDA: 11-JAN-13
IDRESERVA: 2
*****
DADES DE LA VENDA AMB EL ID 3
*****
IDVENDA: 3
.....
.....
```

5.3.7.-Llistat de revisions

Per a consultar el llistat de totes les revisions que hi ha en el sistema s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.LLISTAT_REVISIONS;
```

El resultat de l'execució és el següent:

```
I*****
LLISTAT DE REVISIONS
*****
DADES DE LA REVISIO AMB EL ID 1
*****
IDCOTXE: 1
DATA: 10-JAN-13
MOTIU: 15000km
ES PRESSUPOST: S
COST: 1035
NOM: MECANIC1      CATEGORIA: PRIMERA  NUM HORES REVISIO: 10      PREU HORES REVISIO: 100
NOM: MECANIC2      CATEGORIA: SEGONA   NUM HORES REVISIO: 1      PREU HORES REVISIO: 20
NOM: MECANIC3MOD    CATEGORIA: TERCERAMOD  NUM HORES REVISIO: 1      PREU HORES REVISIO: 15
*****
DADES DE LA REVISIO AMB EL ID 2
*****
IDCOTXE: 2
.....
.....
```

5.3.8.-Llistat de reparacions

Per a consultar el llistat de totes les reparacions que hi ha en el sistema s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.LLISTAT_REPARACIONS;
```

El resultat de l'execució és el següent:

```
*****
LLISTAT DE REPARACIONS
*****
DADES DE LA REPARACIO AMB EL ID 1
*****
IDCOTXE: 1
DATA INICI REPARACIO: 05-OCT-12
DATA FI REPARACIO: 17-OCT-12
MOTIU: Canvi Embragament
COST PECES REPARACIO: 1000
ES PRESSUPOST: N
COST TOTAL: 2035
NOM: MECANIC1      CATEGORIA: PRIMERA  NUM HORES REVISIO: 10      PREU HORES REVISIO: 100
NOM: MECANIC2      CATEGORIA: SEGONA   NUM HORES REVISIO: 1      PREU HORES REVISIO: 20
NOM: MECANIC3MOD    CATEGORIA: TERCERAMOD  NUM HORES REVISIO: 1      PREU HORES REVISIO: 15
*****
DADES DE LA REPARACIO AMB EL ID 2
*****
IDCOTXE: 2
DATA INICI REPARACIO: 06-NOV-12
DATA FI REPARACIO: 18-NOV-12
MOTIU: Problemes caixa canvis
COST PECES REPARACIO: 2000
ES PRESSUPOST: N
COST TOTAL: 2130
NOM: MECANIC1      CATEGORIA: PRIMERA  NUM HORES REVISIO: 2      PREU HORES REVISIO: 30
NOM: MECANIC2      CATEGORIA: SEGONA   NUM HORES REVISIO: 2      PREU HORES REVISIO: 20
NOM: MECANIC3MOD    CATEGORIA: TERCERAMOD  NUM HORES REVISIO: 2      PREU HORES REVISIO: 15
*****
DADES DE LA REPARACIO AMB EL ID 3
*****
IDCOTXE: 21
.....
.....
```

5.3.9.-Avisar revisions per data

Per a simular que s'avisen a clients que han de realitzar una revisió en un rang de data concret, s'ha d'executar el següent procediment:

```
EXEC PKG_CONSULTES.AVISAREVISIONSDATA(TO_DATE('20130101','YYYYMMDD'),TO_DATE('20130331','YYYYMMDD'));
```

El resultat de l'execució és el següent:

```
*****  
AVISANT PER REVISIÓ AMB ID 1  
*****  
DATA REVISIO: 10-JAN-13  
NOM CLIENT: CLIENT1F  
TELEFON CLIENT: 666333111  
*****  
AVISANT PER REVISIÓ AMB ID 2  
*****  
DATA REVISIO: 07-FEB-13  
NOM CLIENT: CLIENT2FMOD  
TELEFON CLIENT: 666333112  
*****  
AVISANT PER REVISIÓ AMB ID 3  
*****  
DATA REVISIO: 08-MAR-13  
NOM CLIENT: CLIENT3F  
TELEFON CLIENT: 666333113
```

5.4.-Mòdul Datawarehouse

En el fitxer *5_4_Proves_ModulDatawarehouse.sql* es troben totes les consultes que apareixen en aquest punt. Si s'executa genera un fitxer de log amb els resultats que apareixen a continuació.

5.4.1.-Inicialització dimensió data

La dimensió data és particular, ja que tots els fets depenen d'ella i no necessita consultar cap dada per inicialitzar-la. Per inicialitzar-la s'executa la següent comanda:

```
EXEC PKG_DWH.INICIALITZA_DIMENSIO_DATA(SYSDATE-2000,SYSDATE+2000);
```

5.4.2.-Càrrega dimensions i fets

Existeix el procediment *ACTUALITZA_DWH* que s'encarrega de carregar totes les dades de les dimensions *client* i *venedor* així com també de tots els fets. Un exemple d'execució és:

```
EXEC PKG_DWH.ACTUALITZA_DWH(to_date('01/01/2011','DD/MM/YYYY'),to_date('31/08/2013','DD/MM/YYYY'));
```

5.4.3.-Temps reparació cotxe

Per a consultar el temps mitjà que es tarda en realitzar una reparació s'ha d'executar el següent procediment:

```
EXEC PKG_DWH.TEMPSMITJANAREPARACIO;
```

El resultat que retorna és:

```
LA MITJANA DE REPARACIO D'UN COTXE ES: 24.21 HORES
```

5.4.4.-Rànquing venedor

Per a consultar el rànkung de venedors s'ha d'executar el següent procediment:

```
EXEC PKG_DWH.RANQUINGMESVENEDOR(2012,10);
```

El resultat que retorna és:

POSICIO 1	IDVENEDOR: 3	QUANTITAT VENUDA: 38000
POSICIO 2	IDVENEDOR: 2	QUANTITAT VENUDA: 22000
POSICIO 3	IDVENEDOR: 4	QUANTITAT VENUDA: 15000
POSICIO 4	IDVENEDOR: 1	QUANTITAT VENUDA: 10000

5.4.5.-Reparacions per mes

Per a consultar el nombre de reparacions que hi ha per mes, s'ha d'executar:

```
EXEC PKG_DWH.RANQUINGMESREPARACIO(201210,201212);
```

El resultat que retorna és:

ANY: 2012	MES: 10	NUM REPARACIONS: 3
ANY: 2012	MES: 11	NUM REPARACIONS: 3
ANY: 2012	MES: 12	NUM REPARACIONS: 3

5.4.6.-Revisions per mes

Per a consultar el nombre de revisions que hi ha per mes s'ha d'executar el següent procediment:

```
EXEC PKG_DWH.RANQUINGMESREVISIO(201211,201212);
```

El resultat que retorna és:

ANY: 2012	MES: 11	NUM REVISIONS: 2
ANY: 2012	MES: 12	NUM REVISIONS: 2

5.4.7.-Vendes per mes

Per a consultar el nombre de vendes que hi ha per mes s'ha d'executar el següent procediment:

```
EXEC PKG_DWH.RANQUINGMESVENDA(201210,201211);
```

El resultat que retorna és:

ANY: 2012	MES: 10	NUM REVISIONS: 4
ANY: 2012	MES: 11	NUM REVISIONS: 5

5.4.8.-Despesa per client

Per a consultar la despesa que ha realitzat cada client i realitzar un rànquing s'ha d'executar el següent procediment:

```
EXEC PKG_DWH.RANQUINGDESPESACLIENT;
```

El resultat que retorna és:

POSICIO: 1	IDCLIENT: 6	DESPESA: 79780
POSICIO: 2	IDCLIENT: 3	DESPESA: 77195
POSICIO: 3	IDCLIENT: 2	DESPESA: 66130
POSICIO: 4	IDCLIENT: 1	DESPESA: 61035
POSICIO: 5	IDCLIENT: 5	DESPESA: 58650
POSICIO: 6	IDCLIENT: 4	DESPESA: 37520
POSICIO: 7	IDCLIENT: 8	DESPESA: 32770
POSICIO: 8	IDCLIENT: 9	DESPESA: 32000
POSICIO: 9	IDCLIENT: 7	DESPESA: 19910

5.5.-Mode Test

En el fitxer *5_5_Proves_ModeTest.sql* es troben totes les consultes que apareixen en aquest punt. Si s'executa genera un fitxer de log amb els resultats de l'execució.

Per a comprovar que el mode test funciona es realitzen una sèrie de passes explicades en els següents punts.

5.5.1.-Recompte dels registres per taula

Per a cada taula (exceptuant les de LOG) es revisa el nombre de registres que té:

```
SELECT 'AGRUPATREPARACIO:'||COUNT(*) FROM AGRUPATREPARACIO;
SELECT 'AGRUPATREVISIO:'||COUNT(*) FROM AGRUPATREVISIO;
SELECT 'AGRUPATVENDA:'||COUNT(*) FROM AGRUPATVENDA;
SELECT 'CATEGORIAMECANIC:'||COUNT(*) FROM CATEGORIAMECANIC;
.....
.....
```

El resultat de l'execució és el següent:

```
CATEGORIAMECANIC:3
AGRUPATREVISIO:8
AGRUPATVENDA:15
CATEGORIAMECANIC:3
.....
.....
```

5.5.2.-Activar el mode Test

Per activar el mode test, s'ha d'executar el següent procediment:

```
EXEC PKG_CONTROL.ACTIVA_TEST;
```

5.5.3.-Càrrega, modificació i esborrada de dades

S'executa els procediments ABM que hi ha en el fitxer *5_2_Proves_ProcedimentsABM.sql*.

5.5.4.-Recompte dels registres per taula

Per a cada taula (exceptuant les de LOG) es revisa de nou el nombre de registres que té:

```
SELECT 'AGRUPATREPARACIO:'||COUNT(*) FROM AGRUPATREPARACIO;
SELECT 'AGRUPATREVISIO:'||COUNT(*) FROM AGRUPATREVISIO;
SELECT 'AGRUPATVENDA:'||COUNT(*) FROM AGRUPATVENDA;
SELECT 'CATEGORIAMECANIC:'||COUNT(*) FROM CATEGORIAMECANIC;
.....
.....
```

El resultat de l'execució és el següent i confirma que no s'han afegit ni esborrat cap registre de les taules del sistema:

```
CATEGORIAMECANIC:3
AGRUPATREVISIO:8
AGRUPATVENDA:15
CATEGORIAMECANIC:3
.....
.....
```

5.6.-Logs

En el fitxer *5_6_Proves_Logs.sql* es troben totes les consultes que apareixen en aquest punt. Si s'executa genera un fitxer de log amb els resultats de l'execució.

Per a comprovar que el LOG funciona es realitzen una sèrie de passes explicades en els següents punts.

5.6.1.-Cerca d'errors

Executant el següent procediment es pot comprovar que no hi ha (s'hi no s'ha provocat) cap error en el LOG del sistema:

```
EXEC PKG_CONTROL.LOG_CERCA_OPERACIO('ERROR');
```

5.6.2.-Provocar error

Executam el següent procediment i provocam un error (ja que existeix un mecànic amb el mateix ID en el sistema):

```
EXEC PKG_RRHH.ALTA_MECANIC(1,'MECANIC1','MECANIC1L1','MECANIC1L2',1);
```

Comprovam que l'error s'ha enregistrat executant el següent procediment de nou:

```
EXEC PKG_CONTROL.LOG_CERCA_OPERACIO('ERROR');
```

El resultat obtingut és el següent:

DADES DEL LOG AMB EL ID 716

PROCEDIMENT LOG: PKG_RRHH.ALTA_MECANIC

PARAMETRES LOG: P1:1 P2:MECANIC1 P3:MECANIC1L1 P4:1

OPERACIO: ERROR ORA-00001: unique constraint (PFCUSER.MECANIC_PK) violated DATA: 11-JAN-13

OPERACIO: INICI DATA: 11-JAN-13

5.6.3.-Desactivar LOG i provocar error

Per a desactivar el LOG s'ha d'executar el següent procediment:

```
EXEC PKG_CONTROL.DESACTIVA_LOG;
```

Executam el següent procediment i provocam de nou un error (ja que existeix un mecànic amb el mateix ID en el sistema):

```
EXEC PKG_RRHH.ALTA_MECANIC(1,'MECANIC1','MECANIC1L1','MECANIC1L2',1);
```

Comprovam que l'error aquest cop no s'ha enregistrat executant el següent procediment de nou:

```
EXEC PKG_CONTROL.LOG_CERCA_OPERACIO('ERROR');
```

El resultat obtingut és el següent (sols hi apareix l'error anterior):

DADES DEL LOG AMB EL ID 716

PROCEDIMENT LOG: PKG_RRHH.ALTA_MECANIC

PARAMETRES LOG: P1:1 P2:MECANIC1 P3:MECANIC1L1 P4:1

OPERACIO: ERROR ORA-00001: unique constraint (PFCUSER.MECANIC_PK) violated DATA: 11-JAN-13

OPERACIO: INICI DATA: 11-JAN-13

5.6.4.-Consultar LOG per ID

Per a consultar un LOG amb un ID concret, s'ha d'executar el següent procediment:
`EXEC PKG_CONTROL.LOG_ID(1);`

El resultat obtingut és el següent:

```
*****
DADES DEL LOG AMB EL ID 1
*****
PROCEDIMENT LOG: PKG_RRHH.ALTA_CATEGORIA_MECANIC
PARAMETRES LOG: P1:0 P2:PRIMERA P2:30
OPERACIO: FI      DATA: 11-JAN-13
OPERACIO: INICI   DATA: 11-JAN-13
```

5.6.5.-Consultar LOG per rang de temps

Per a consultar el detall de LOG en un rang de temps (en aquest cas els darrers 5 segons), s'ha d'executar el següent procediment:

`EXEC PKG_CONTROL.LOG_RANG_DATA(SYSDATE-5/86400,SYSDATE);`

El resultat obtingut és el següent:

```
*****
DADES DEL LOG AMB EL ID 794
*****
PROCEDIMENT LOG: PKG_CONTROL.LOG_CERCA_OPERACIO
PARAMETRES LOG: P1:ERROR
OPERACIO: FI      DATA: 12-JAN-13
OPERACIO: INICI   DATA: 12-JAN-13
*****
DADES DEL LOG AMB EL ID 795
*****
PROCEDIMENT LOG: PKG_CONTROL.LOG_ID
PARAMETRES LOG: P1:793
OPERACIO: FI      DATA: 12-JAN-13
OPERACIO: INICI   DATA: 12-JAN-13
*****
DADES DEL LOG AMB EL ID 796
*****
PROCEDIMENT LOG: PKG_CONTROL.LOG_ID
PARAMETRES LOG: P1:716
OPERACIO: FI      DATA: 12-JAN-13
OPERACIO: INICI   DATA: 12-JAN-13
```

6.-Valoració econòmica

El nombre total d'hores dedicades al projecte són 279. Agrupant les hores segons el perfil, la valoració econòmica del mateix és la següent:

PERFIL	Hores	Preu Hora	Total
Cap de projecte	39.5	55.00 €	2,172.50 €
Analista	74.5	45.00 €	3,352.50 €
Programador	51.5	35.00 €	1,802.50 €
Tester	41.5	30.00 €	1,245.00 €
Documentalista	72	20.00 €	1,440.00 €
Total	279	185.00 €	10,012.50 €

És a dir, el pressupost del projecte pel que fa a mà d'obra (sense comptar amb les llicències d'Oracle ni del programari utilitzat) és d'uns 10.012,50€.

7.-Conclusions

La principal conclusió que podem extreure d'aquest projecte és que una bona planificació és fonamental per a finalitzar correctament un projecte. Un exemple ho trobam en l'anàlisi de riscos i en la definició d'un pla de contingència. Això facilita que en el cas de trobar-mos davant un risc contemplat, sapiguem com solucionar-lo de forma ràpida.

La comunicació amb el client és un punt important en un projecte de programari. Si aquesta no és fluida i clara pot provocar, per exemple, que la definició dels requeriments no sigui l'adequada. Pot arribar a succeir que un cop finalitzat el projecte ens adonem que el que s'ha implementat no satisfà les necessitats del client i, per tant, s'hagin de realitzar més hores de les planificades, augmentant els costos del projecte i el temps d'espera per part del client.

Una altra conclusió important que se n'extreu és que una documentació clara i senzilla facilita la realització del projecte. Un exemple el trobam en la definició dels casos d'ús. Si l'analista no explica de forma clara els casos d'ús que s'han d'implementar, com al punt anterior, pot fer que el desenvolupador no implementi tot el que s'hauria d'implementar i que no satisfacin les necessitats del client.

Pel que fa a les fases del projecte, totes les he trobades interessants i més o manco han estat igual de difícils. Destacar que la que m'ha costat una mica més ha estat la darrera (més orientada a la documentació). A algunes assignatures de la carrera ja s'indicava que a l'hora de desenvolupar un dels punts més importants (i que més costa realitzar) és el de la documentació. En aquest projecte ho he pogut certificar.

Destacar també que considero molt interessant el fet que s'ens faci fer una valoració econòmica del projecte. Així (de forma orientativa) es reconeix des del punt de vista econòmic el valor que pot arribar a tenir un projecte de programari, punt de vista que molt de cops no es sol tenir en compte en la informàtica.

Finalment, des del punt de vista personal considero que la realització d'aquest projecte no s'ha de tenir en compte sols com a un exercici didàctic, si no que també s'ha de veure com a una gran experiència pràctica dins del món del desenvolupament de programari en BD. Això pot obrir-me portes a l'hora d'aconseguir un lloc de treball en empreses on hagi de desenvolupar una aplicació o sistema en qualsevol SGBD.

8.-Glossari

Clau forana: és un atribut d'una taula que fa referència a un atribut d'una altra taula.

Clau primària: és un atribut que identifica una entitat i que té un valor únic. No hi pot haver dues files dins una taula amb una clau primària iguala.

Entitat: és una representació abstracte de qualque cosa del món real que té una sèrie de propietats que es poden representar mitjançant atributs.

Relació: és una associació entre entitats.

Diagrama E/R: és una eina utilitzada per al modelat de les dades d'un sistema d'informació relacional. Aquest model expressa entitats rellevants d'un sistema, les seves propietats i les interrelacions entre elles.

SGBD: sigles de Sistema de Gestió de Base de Dades.

Procediment emmagatzemat: són programes emmagatzemats i compilats en la BD i que quan es criden s'executen directament en el SGBD. Això permet manipular grans quantitats de dades entre un client i el servidor on s'executa el SGBD sense saturar la xarxa.

Paquet: un paquet és una estructura que agrupa objectes de PL/SQL compilats en una base de dades. Permet agrupar les funcionalitats dels processos dels programes.

Script: és un fitxer de text que conté les instruccions necessàries per a realitzar qualque tipus de tasca. En aquest projecte s'han utilitzat tant per crear la BD, com un usuari, com per crear tots els paquets o taules del sistema implementat.

Procediments ABM: procediments emmagatzemats que permeten realitzar altes, baixes i modificacions de dades en el sistema.

PFC: projecte de final de carrera.

9.-Bibliografia

9.1.-Documentació

- Bases de dades: Universitat Oberta de Catalunya
- Bases de dades II: Universitat Oberta de Catalunya
- Metodologia i Gestió de Projectes TIC: Universitat Oberta de Catalunya

9.2.-Enllaços d'internet

- <http://meandmyubuntulinux.blogspot.com.es/2012/05/installing-oracle-11g-r2-express.html>
- <http://www.google.com/>
- http://docs.oracle.com/cd/B28359_01/appdev.111/b28843/toc.htm

10.-Annex

Manual d'instal·lació i proves

Les passes que s'han de realitzar per a instal·lar i després provar el sistema que s'ha implementat són:

1. Instal·lar el SGBD amb el qual s'ha de treballar. En aquest cas és el ORACLE Express 11gR2 (si s'utilitza el sistema operatiu Ubuntu convé seguir els passos de la plana web <http://meandmyubuntulinux.blogspot.com.es/2012/05/installing-oracle-11g-r2-express.html>)
2. Un cop instal·lat el SGBD es poden utilitzar eines gràfiques (en aquest cas el *SQLDeveloper*) que facilitaran l'execució de scripts o es poden utilitzar directament eines proporcionades pel propi SGBD com poden ser el *SQL PLUS*.
3. Una vegada s'ha instal·lat el SGBD, si no s'ha creat la BD (molt probable si s'instal·la sobre Ubuntu), convé executar el script *3_3_1_DissenyFisic_CreacioBD.sql*.
4. Un cop creada la BD, ens hem de connectar com a usuari administrador (*sys o system*) i creem el tablespace executant el script *3_3_2_DissenFisic_CreacioTablespace.sql*. És probable que s'hagi de modificar la ubicació del tablespace si la instal·lació del SGBD es realitza sobre un sistema operatiu diferent al Ubuntu.
5. La següent passa és crear l'usuari de connexió del sistema. Per fer-ho s'ha d'executar el script *3_3_3_DissenyFisic_CreacioUsuari*.
6. En aquest punt ja es pot instal·lar el codi del sistema. Per fer-ho el més senzill és executar el script *CrearCodi.sql*. Aquest s'encarrega d'executar els scripts *3_3_4_DissenyFisic_CreacioTaules.sql*, *3_3_6_DissenyFisic_Sequencies.sql* i *4_Implementacio.sql*, el qual a l'hora executa tots els scripts que comencen per *4_?_Implementacio_*.sql*.
7. Un cop està tot implementat, si es volen executar totes les proves que s'han implementat (incloent el poblar el sistema), és suficient executant el script *5_Proves.sql*. Si el que es vol, en canvi, és tenir el sistema buit però preparat per utilitzar-se és suficient executant el script *5_1_Proves_CarregaInicial.sql*, el qual insereix una sèrie de registres indispensables en unes taules per poder utilitzar el sistema.