

TFC
Bases de dades relacionals
Enginyeria tècnica en Informàtica de Gestió

**Disseny i implementació d'un sistema de control
d'un consultori mèdic**

Memòria

M. Rosa Pradera Villegas
Consultor: Alexandre Cornet
Setembre 2005 – Gener 2006

ÍNDEX

ÍNDEX.....	2
1.- RESUM.....	3
2.- INTRODUCCIÓ	4
2.1.- DESCRIPCIÓ I ABAST DEL TFC	4
2.2.- OBJECTIUS GENERALS I ESPECÍFICS.....	7
2.3.- PLANIFICACIÓ AMB FITES I TEMPORALITZACIÓ	8
2.3.1.- DESCRIPCIÓ DE TASQUES I FITES	8
2.3.2.- PLANIFICACIÓ TEMPORAL	10
2.3.3.- FITES DEL PROJECTE.....	12
3.- ANÀLISI.....	13
3.1.- REQUISITS FUNCIONALS	13
3.2.- REQUISITS NO FUNCIONALS	14
4.- DISSENY	16
4.1.- DISSENY CONCEPTUAL	16
4.1.1.- ENTITATS I INTERRELACIONS	16
4.1.2.- DIAGRAMA ER.....	21
4.2.- DISSENY LÒGIC.....	22
4.2.1.- DIAGRAMA ER FINAL.....	23
4.2.2.- RESTRICCIONS D'INTEGRITAT	24
4.2.3.- REGLES DE NEGOCI	24
4.3.- DISSENY FÍSIC	25
4.3.1. RELACIONS.....	25
4.3.2. DISPARADORS	29
4.3.3. PROCEDIMENTS EMMAGATZEMATS.....	34
4.3.4. ÍNDEXS SECUNDARIS	40
4.3.5. USUARIS	40
5.- IMPLEMENTACIÓ.....	42
6.- PROVES	42
7.- CONCLUSIONS.....	43
8.- BIBLIOGRAFIA.....	44

1.- RESUM

Aquest treball consisteix en el disseny i implementació d'un sistema informàtic per a la gestió d'un consultori mèdic, amb les següents funcionalitats: gestió de pacients, control d'hores de visita, gestió de medicaments i manteniment d'historials mèdics.

Després de l'anàlisi dels requeriments, es realitza el disseny de la base de dades partint del model conceptual emprant el Diagrama E/R. Posteriorment s'implementa utilitzant el sistema de gestió de bases de dades Oracle 9i.

Es necessari també crear i implementar procediments per a la manipulació de les dades de la base de dades, els quals han de complir una sèrie de requisits.

Finalment s'elabora un joc de proves per comprovar el bon funcionament del sistema.

2.- INTRODUCCIÓ

2.1.- DESCRIPCIÓ I ABAST DEL TFC

Actualment, la importància que té la informació en totes les organitzacions ha fet imprescindible disposar de bases de dades que permeti representar-la i manipular-la adequadament.

Les bases de dades relacionals ens proporcionen sistemes per representar i emmagatzemar tot tipus de dades, i manipular-les sempre respectant-ne la integritat (condicions que les dades han de complir). La base de dades es vista per l'usuari com una estructura lògica consistent en un conjunt de relacions.

Aquest Treball fi de carrera es troba dins l'àrea de les bases de dades. Consisteix en realitzar el disseny d'un sistema informàtic capaç de dur part de la gestió d'un consultori mèdic. Aquest sistema disposarà, com a mínim, de les següents funcionalitats:

- Gestió de pacients: s'han de poder realitzar altes, baixes i modificacions d'aquests.
- Gestió d'hores de visita: cada una disposarà d'un temps mínim que vindrà donat per un paràmetre de configuració emmagatzemat en una taula de paràmetres del sistema. Cada visita tindrà associada una fitxa que contindrà, entre d'altres dades, el diagnòstic, anotacions realitzades pel metge durant la visita i els medicaments receptats.
- Gestió de medicaments: cada medicament tindrà associades les seves característiques, marca i família a la qual pertanyen.
- Manteniment d'historials mèdics.

Després de l'anàlisi, per realitzar el disseny de la base de dades partirem del model conceptual emprant el Diagrama E/R. En la etapa d'implementació caldrà crear els *scripts* (codi que el sistema de gestió de bases de dades pot interpretar) de creació de les estructures de la base de dades i els procediments necessaris.

Les taules on s'emmagatzemaran les dades tindran permís només de lectura, la qual cosa implica que si l'usuari vol accedir a la base de dades per fer alguna modificació (introduir, modificar o eliminar alguna dada) ho haurà de fer mitjançant procediments emmagatzemats. Un procediment és una unitat de codi que fa una tasca, pot rebre arguments i retornar valors.

Així, caldrà crear procediments per donar d'alta, modificar i esborrar els pacients, les hores de visita, les fitxes, els medicaments i els historials mèdics. Es tindrà en compte que alguns estaran relacionats, per exemple esborrar un pacient implica un esborrat en cascada, ja que pot tenir assignada una hora de visita i/o un historial mèdic.

Tots els procediments emmagatzemats hauran de complir uns requisits:

- com a mínim tindran un paràmetre de sortida anomenat RSP, de tipus *string* (tipus text), que contindrà valor 'OK' si l'execució ha finalitzat amb èxit, en cas contrari indicarà el valor i el tipus d'error.
- tractament d'excepcions. Una excepció és una situació que interromp l'execució normal del programa deguda a un error d'aquest o a un esdeveniment inesperat. S'haurà de gestionar adequadament perquè o continuï l'execució (si és possible) o es notifiqui de l'error i es mantingui la integritat i consistència de les dades. L'error es notificarà en termes comprensibles per a l'usuari.

Totes les crides a procediments s'emmagatzemaran en un taula de *log*, les dades seran: el nom del procediment executat, els paràmetres d'entrada, els de sortida, el dia i l'hora. Això ens permetrà fer un seguiment de la utilització i funcionament de la base de dades, així com obtenir estadístiques que poden ser d'utilitat (per exemple podríem saber quin dia de la setmana és recepten més analgèsics).

Es farà ús de traces dins el codi per facilitar-ne el seguiment i la depuració d'aquest. Així, si el programa no funciona correctament es pot saber en quina part concreta succeeix l'error. També es crearà un Joc de Proves per comprovar el bon funcionament.

El sistema de gestió de bases de dades utilitzat per a la part pràctica del treball serà Oracle.

Abast del projecte

El fet de dissenyar una base de dades que hagi de contenir historials mèdics té unes implicacions legals que en un cas real haurien de ser contemplades. Per exemple, el temps que s'ha de mantenir una història clínica abans de ser donada de baixa, l'accés a les dades i la modificació del contingut.

Tot i que tenir en compte aquests aspectes comporta afegir una mica més de complicació, i que no és necessari, s'ha decidit fer-ho per aproximar aquest projecte el màxim possible a la realitat amb la que ens podríem trobar en el futur.

Per tant, es tindrà en compte que:

- una història mèdica no es pot eliminar abans de cinc anys de la data d'alta del seu procés assistencial (*BOE, LEY 41/2002 de 14 de noviembre*). Cal doncs afegir aquesta condició dintre del procediment per esborrar historials mèdics.
- l'accés al contingut de la història quedarà limitat als professionals sanitaris que intervinguin en el seu procés assistencial (metge, psicòleg, infermer, fisioterapeuta, etc). Això implicarà gestionar adequadament els usuaris, assignant-los un perfil, i donar permisos sobre determinades dades segons el perfil d'usuari. Per exemple, l'administratiu que assigna les hores de visita podrà veure les dades del pacient, però no el contingut del historial mèdic.

- el contingut d'una història no s'ha de poder modificar, si no és per afegir-hi informació (en cas de ser necessari corregir algun error, s'afegirien les anotacions oportunes indicant que es pretén corregir el que s'hi havia posat anteriorment). Es a dir, el procediment per modificar una història crearia una fitxa nova on poder deixar constància de la modificació.

2.2.- OBJECTIUS GENERALS I ESPECÍFICS

Objectius generals

L'objectiu general del TFC és posar en pràctica i ampliar els coneixements adquirits durant els estudis, especialment en l'àrea de les bases de dades, mitjançant el desenvolupament d'un projecte informàtic.

Objectius específics

- Posar en pràctica els coneixements adquirits en les assignatures de Bases de dades I i Bases de dades II.
- Conèixer i treballar amb un sistema de gestió de bases de dades que no s'ha utilitzat durant els estudis, concretament amb Oracle.
- Profunditzar en el control d'errors i tractament d'excepcions.
- Emprar el llenguatge PL/SQL.

2.3.- PLANIFICACIÓ AMB FITES I TEMPORALITZACIÓ

2.3.1.- DESCRIPCIÓ DE TASQUES I FITES

- 1.**PLANIFICACIÓ**: es defineix i planifica el projecte.
- 2.**TROBADA D'INICI**: s'expliquen les bases del projecte.
- 3.**DESCRIURE EL TFC**: es descriu el projecte.
- 4.**DEFINIR OBJECTIUS**: es defineixen els objectius del projecte.
- 5.**DEFINIR TASQUES**: s'identifiquen i defineixen les tasques.
- 6.**PLANIFICACIÓ TEMPORAL**: s'assigna temps a cada tasca.
- 7.**ELABORAR PLA DE TREBALL**: s'elabora el pla de treball i la documentació corresponent.

- 8.**LLIURAR PAC 1**: es lliura el Pla de Treball al consultor.

- 9.**INSTAL·LAR PROGRAMARI**: s'instal·la el programari necessari per al projecte.
- 10.**INSTAL·LAR ORACLE**: s'instal·la el motor de bases de dades Oracle.
- 11.**INSTAL·LAR PL/SQL DEVELOPER**: s'instal·la l'entorn de programació PL/SQL Developer.
- 12.**REDACTAR DOCUMENTACIÓ DEL PROGRAMARI**: es redacta la documentació sobre la instal·lació del programari.

- 13.**ESTUDIAR ORACLE I PL/SQL**: s'estudia el funcionament de Oracle i el llenguatge de programació PL/SQL.

- 14.**ETAPA D'ANÀLISI**: s'analitzen els requeriments dels usuaris i de les àrees d'aplicació.
- 15.**ESPECIFICAR REQUISITS**: s'obté un conjunt d'especificacions de requisits (funcionals, no funcionals, etc.)
- 16.**REDACTAR DOCUMENTACIÓ**: es redacta la documentació corresponent a l'etapa d'anàlisi.

- 17.**ETAPA DE DISSENY**: es dissenya l'estructura de la base de dades.
- 18.**DISSENY CONCEPTUAL**: s'obté l'estructura de la base de dades independentment de la tecnologia i del tipus (relacional, orientada a objectes, etc.):
 - s'identifiquen les entitats (objectes o conceptes dels que es recull informació), les relacions i els atributs (característiques de les entitats i les relacions),
 - es determinen els identificadors (un o més atributs que identifiquen de manera única cada entitat o relació),
 - es determinen les jerarquies (si n'hi ha),
 - es fa el diagrama ER (entitat – interrelació).

- 19.**DISSENY LÒGIC**: s'ajusta el disseny conceptual al model relacional.
 - s'obtenen un conjunt de relacions amb els seus atributs, claus primàries (identificadores) i claus foranes (que relacionen amb una altra entitat o relació),
 - es defineixen les restriccions d'integritat (regles que la base de dades ha de complir).

20.DISSENY FÍSIC: es tradueix l'esquema lògic en un esquema que es pugui implementar en Oracle, incloent les estructures d'emmagatzematge i mètodes d'accés:

- es dissenyen les relacions i els atributs en llenguatge propi de Oracle,
- es dissenyen les regles de negoci per a Oracle (restriccions que han de complir les operacions sobre la base de dades),
- es dissenyen els procediments per crear, modificar i esborrar pacients, hores de visita, les fitxes, els medicaments i els historials mèdics.
- s'escullen els índex secundaris (per exemple per a aquells atributs mitjançant els quals s'accedeix freqüentment a la base de dades),
- s'estima la necessitat d'espai,
- es dissenyen les vistes que tindran els usuaris,
- es dissenyen les regles d'accés (els permisos que els diferents usuaris tindran sobre les dades).

21.REDACTAR DOCUMENTACIÓ: es redacta la documentació corresponent al disseny.

22.LLIURAR PAC 2: es lliura la PAC 2 al consultor.

23.ETAPA D'IMPLEMENTACIÓ: s'implementa el disseny físic.

24.CREAR CODI DE GENERACIÓ BD: es crea el codi que generarà la base de dades.

25.CREAR CODI D'ACCÉS A LA BD: es crea el codi per accedir a la base de dades.

26.REDACTAR DOCUMENTACIÓ: es redacta la documentació corresponent a la implementació.

27.PROVES: es comprova el funcionament de la base de dades.

28.CREAR PROVES: es creen les proves necessàries i es comparen els resultats obtinguts amb els esperats.

29.MODIFICAR BD SI S'ESCAU: es modifica allò que no funciona correctament.

30.REDACTAR DOCUMENTACIÓ: es redacta la documentació corresponent al Joc de Proves.

31.LLIURAR PAC 3: es lliura la PAC 3 al consultor (documentació i codi).

32.TROBADA DE SÍNTESI: es comenta el projecte amb el consultor.

33.PRESENTACIÓ I MEMÒRIA: es fa el redactat definitiu de la Memòria i es presenta el TFC.

34.REDACTAR CONCLUSIONS: s'elaboren les conclusions del projecte.

35.PRESENTACIÓ PROVISIONAL: es fa una presentació provisional del projecte.

36.CREACIÓ PRESENTACIÓ POWERPOINT: s'elabora la presentació virtual del projecte.

37.REVISIONS: es fan les revisions oportunes sobre la presentació provisional.

38.ENTREGAR TFC (MEMÒRIA, PRESENTACIÓ I CODI): es lliuren la memòria, la presentació i el codi definitius.

39.DEBAT: es debat el TFC amb el Tribunal.

2.3.2.- PLANIFICACIÓ TEMPORAL

El quadre següent mostra la planificació temporal del projecte, realitzada amb Microsoft Office Project. El calendari laboral aplicat és de 3 hores diàries de dilluns a diumenge i la durada prevista és de 126 dies.

	Nombre de tarea	Duración	Comienzo	Fin
1	Planificació	8,5 días	sáb 17/09/05	dom 25/09/05
2	Trobada d'inici	1 día	sáb 17/09/05	sáb 17/09/05
3	Descriure el TFC	1,5 días	dom 18/09/05	lun 19/09/05
4	Definir objectius	0,5 días	lun 19/09/05	lun 19/09/05
5	Definir tasques	1,5 días	mar 20/09/05	mié 21/09/05
6	Planificació temporal	2 días	mié 21/09/05	vie 23/09/05
7	Elaborar Pla de Treball	1,5 días	sáb 24/09/05	dom 25/09/05
8	Lliurar PAC 1	0 días	lun 26/09/05	lun 26/09/05
9	Instal·lar programari	2 días	mar 27/09/05	mié 28/09/05
10	Instal·lar Oracle	0,5 días	mar 27/09/05	mar 27/09/05
11	Instal·lar PL/SQL Developer	0,5 días	mar 27/09/05	mar 27/09/05
12	Redactar documentació del programari	1 día	mié 28/09/05	mié 28/09/05
13	Estudiar Oracle i PL/SQL	20 días	mar 27/09/05	dom 16/10/05
14	Etapa d'Anàlisi	3 días	dom 25/09/05	mié 28/09/05
15	Especificar requisits	2 días	dom 25/09/05	mar 27/09/05
16	Redactar documentació	1 día	mar 27/09/05	mié 28/09/05
17	Etapa de Disseny	28 días	mié 28/09/05	mié 26/10/05
18	Disseny Conceptual (fer diagrama ER)	3 días	mié 28/09/05	sáb 01/10/05
19	Disseny Lògic (transformar ER a lògic)	1 día	sáb 01/10/05	dom 02/10/05
20	Disseny Físic (implementable en Oracle)	20 días	dom 02/10/05	sáb 22/10/05
21	Redactar documentació	4 días	sáb 22/10/05	mié 26/10/05
22	Lliurar PAC 2	0 días	mié 02/11/05	mié 02/11/05
23	Etapa d'Implementació	26 días	mié 26/10/05	lun 21/11/05
24	Crear codi de generació BD	4 días	mié 26/10/05	dom 30/10/05
25	Crear codi d'accés a la BD	20 días	dom 30/10/05	sáb 19/11/05
26	Redactar documentació	2 días	sáb 19/11/05	lun 21/11/05
27	Proves	18 días	lun 21/11/05	vie 09/12/05
28	Crear proves	15 días	lun 21/11/05	mar 06/12/05
29	Modificar BD si s'escau	15 días	lun 21/11/05	mar 06/12/05
30	Redactar documentació	3 días	mar 06/12/05	vie 09/12/05
31	Lliurar PAC 3	0 días	lun 12/12/05	lun 12/12/05
32	Trobada de síntesi	1 día	sáb 17/12/05	sáb 17/12/05
33	Presentació i Memòria	19 días	vie 09/12/05	mié 28/12/05
34	Redactar conclusions	2 días	vie 09/12/05	dom 11/12/05
35	Presentació provisional	7 días	dom 11/12/05	dom 18/12/05
36	Creació presentació PowerPoint	3 días	dom 18/12/05	mié 21/12/05
37	Revisions	10 días	dom 18/12/05	mié 28/12/05
38	Entregar TFC (Memòria, presentació i codi)	0 días	lun 09/01/06	lun 09/01/06
39	Debat	5 días	lun 16/01/06	vie 20/01/06

Diagrama de Gantt

El diagrama de Gantt és una representació gràfica de l'ordre i la distribució temporal de les tasques.

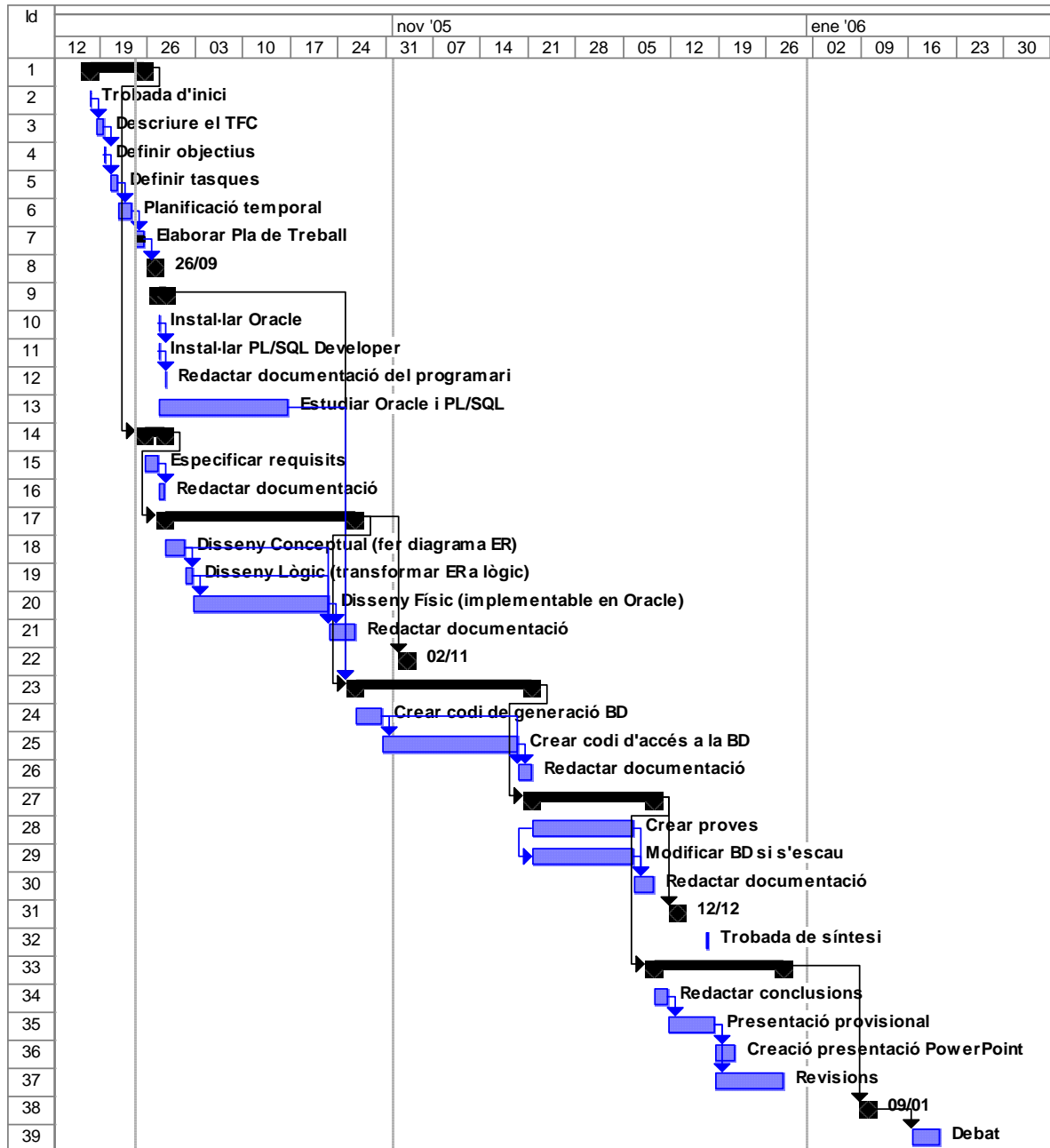
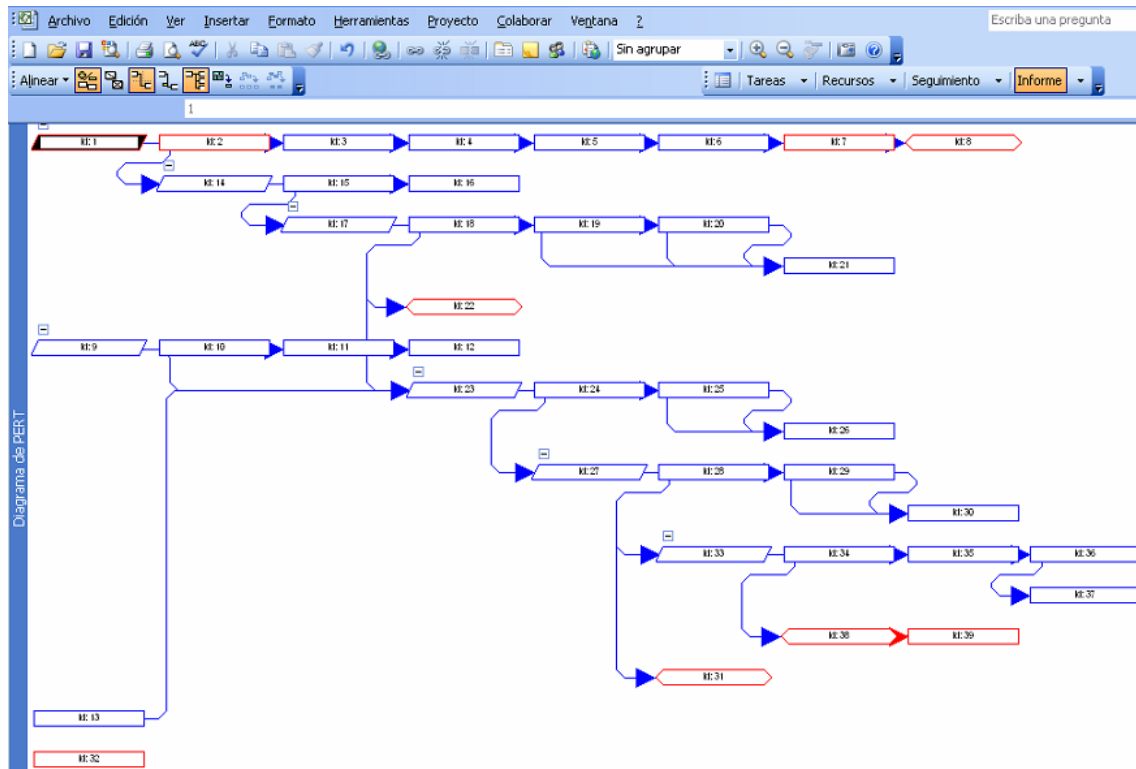


Diagrama de PERT

El diagrama de PERT permet veure les tasques que es poden moure en el temps sense afectar el temps de finalització del projecte, i les tasques crítiques, les quals podrien causar retard en el temps total.



2.3.3.- FITES DEL PROJECTE

Nom		Data
PAC 1	Pla de treball	26/09/05
PAC 2	Anàlisi i Disseny	02/11/05
PAC 3	Implementació	12/12/05
TFC	Memòria, codi, presentació virtual	09/01/06

3.- ANÀLISI

En aquesta etapa s'analitzen els requeriments dels usuaris i de les àrees d'aplicació. S'obtindran un conjunt d'especificacions de requisits.

3.1.- REQUISITS FUNCIONALS

Gestió de pacients

De cada pacient tindrem les seves dades personals: DNI, nom i cognoms, data de naixement, adreça i telèfon. Si es tractés d'un consultori de la sanitat pública, seria necessari afegir el número de la Seguretat Social i/o targeta sanitària.

Els pacients han de poder ser donats d'alta, però no podran ser donats de baixa mentre tinguin una història clínica associada.

Només es podrà modificar l'adreça, el telèfon o afegir el DNI si no en tenia.

Aquestes dades podran ser consultades per tots els usuaris de la base de dades.

Control d'hores de visita

De cada hora de visita tindrem: dia, hora, un temps mínim (que vindrà donat per un paràmetre de configuració emmagatzemat en una taula de paràmetres del sistema) i el pacient.

Podem tenir més d'un metge o professional sanitari. Per tant, per a cada hora de visita haurem de tenir també un professional.

Una vegada creades les corresponents hores de visita per a cada professional, es podrà assignar o anul·lar un pacient a aquestes.

Les hores de visita podran ser consultades per tots els usuaris. Els administratius seran els encarregats d'assignar els pacients. I l'administrador del consultori les podrà crear (en funció de l'horari del consultori, per a aquest cas serà de 9 a 13 h.) i eliminar.

Gestió de medicaments

De cada medicament tindrem: el nom genèric, noms comercials, i grup al qual pertany. No es pot eliminar un medicament si ha estat receptat a algun pacient.

Els medicaments podran ser donats d'alta i de baixa per l'administrador del consultori, i consultats per tots els professionals sanitaris. Només podran ser receptats, però, pel metge que fa la visita.

Manteniment d'historials mèdics

Cada historial contindrà la informació del pacient i de les visites. A cada visita que faci el pacient al centre se li associarà una fitxa que contindrà el diagnòstic, anotacions realitzades pel metge durant la visita i els medicaments receptats. Interessa saber també si es dona d'alta al pacient del seu procés assistencial.

Les fitxes podran ser donades d'alta i consultades només per professionals sanitaris. En cap cas han de poder ser modificades, si es necessita fer-ho es donarà d'alta una altra per introduir les modificacions oportunes. Només els professionals sanitaris podran donar d'alta una fitxa i consultar-la.

La seva eliminació vindrà condicionada a la eliminació completa de la història clínica de la qual formen part. Es a dir, només es poden esborrar si es fa en conjunt (el pacient i totes les seves fitxes), i si han passat com a mínim 5 anys de la data d'alta de l'últim procés assistencial, tal com marca la legislació. L'esborrat d'un historial mèdic anirà a càrrec de l'administrador del consultori.

Taula de log

Tots els accessos a la base de dades, excepte la lectura, quedaran enregistrats en una taula. S'emmagatzemaran: el nom del procediment executat, el dia i l'hora, l'usuari, els paràmetres d'entrada i els de sortida.

3.2.- REQUISITS NO FUNCIONALS

Interfície d'usuari

El disseny d'aquest sistema no contempla el d'una interfície gràfica per a l'usuari final del producte, restarà doncs pendent de desenvolupar.

Per realitzar els accessos a la base de dades s'utilitzarà PL/SQL Developer (és un entorn que permet interactuar amb la base de dades fàcilment), ja sigui mitjançant la clàusula *SELECT* per visualitzar continguts, o utilitzant els procediments creats per a la resta de funcionalitats.

Comportament

El sistema haurà de tenir un comportament estable, informant a l'usuari si l'execució ha finalitzat amb èxit, en cas contrari informarà adequadament de l'error.

Tots els accessos a la base de dades mitjançant procediments quedaran enregistrats en una taula de *log*, on tindrem: dia i hora, nom del procediment, els paràmetres d'entrada i els de sortida.

Volums d'informació

No sabem el volum de pacients del consultori, la quantitat de visites previstes per pacient, ni la quantitat de metges, etc. No es demana, per a aquest projecte, tenir en compte limitacions d'espai ni com dimensionar el servidor. Si fos un cas real, però, caldria fer-ho.

Usuaris

Tindrem quatre perfils d'usuari:

- Administrador de la base de dades: amb permisos il·limitats sobre la base de dades.
- Administrador del consultori:
 - amb tasques administratives,
 - inserirà i esborrarà les hores de visita,
 - inserirà i esborrarà els professionals,
 - inserirà i esborrarà els medicaments,
 - esborrarà els pacients i els seus historials, però no podrà veure el contingut de les fitxes.
- Professional sanitari (metge, infermer, psicòleg, etc):
 - crearà les fitxes corresponents a les visites,
 - podrà fer receptes si és metge.
- Administratiu:
 - inserirà pacients a la base de dades,
 - assignarà o anul·larà les hores de visita als pacients,
 - no podrà veure el contingut de les fitxes.

Programari

El sistema gestor de bases de dades serà Oracle i l'accés es farà utilitzant PL/SQL Developer.

4.- DISSENY

4.1.- DISSENY CONCEPTUAL

En aquesta etapa s'obté una estructura de la base de dades independentment de la tecnologia i el tipus. S'identifiquen les entitats, les interrelacions i els atributs.

4.1.1.- ENTITATS I INTERRELACIONS

1.- USUARI

Guarda tots els usuaris.

Clau primària formada per: **id_usuari**.

id_usuari

Camp alfanumèric de longitud 6. No pot ser nul.
Identifica a l'usuari.

contrasenya

Camp alfanumèric de longitud 6.
Contrasenya de l'usuari.
No pot ser nul·la, per defecte serà id_usuari.

rol

Camp alfanumèric de longitud 30.
Identifica el rol de l'usuari. No pot ser nul.

2.- ROL

Guarda els diferents rols d'usuari.

Clau primària formada per: **tipus**.

tipus

Camp alfanumèric de longitud 30.
Identifica el rol: administrador de la base de dades, administrador del consultori, professional sanitari, administratiu, etc. No pot ser nul.

3.- ÉS

Interrelació entre USUARI i ROL.

Un usuari pot tenir més d'un rol.

Un rol el pot tenir més d'un usuari.

Tot usuari té un rol.

Pot haver-hi rols que no els tingui cap usuari.

4.- PACIENT

Guarda tots els pacients.

Clau primària formada per: **numHistorial**.

numHistorial

Camp numèric incrementable automàticament de longitud 6.
Número de historial del pacient. No pot ser nul.

dni

Camp numèric de longitud 8.
Identifica el DNI. Pot ser nul, per exemple un menor que no tingui DNI.

nom

Camp alfanumèric de longitud 20.
Nom del pacient. No pot ser nul.

cognom1

Camp alfanumèric de longitud 20.
Primer cognom del pacient. No pot ser nul.

cognom2

Camp alfanumèric de longitud 20.
Segon cognom del pacient. No pot ser nul, en el cas d'una persona amb només un cognom es repetiria el primer.

dataNaixement

Camp amb format data.
Data de naixement del pacient. No pot ser nul·la.

adreça

Camp alfanumèric de longitud 50.
Adreça del pacient. No pot ser nul·la.

telefon

Camp numèric de longitud 9.
Número de telèfon del pacient. Pot ser nul.

5.- PROFESSIONAL

Guarda tots els professionals del consultori mèdic.
Clau primària formada per: **id_professional**.

id_professional

Camp numèric incrementable automàticament de longitud 4.
Identifica el professional. No pot ser nul.

nom

Camp alfanumèric de longitud 20.
Nom del pacient. No pot ser nul.

cognom1

Camp alfanumèric de longitud 20.
Primer cognom del pacient. No pot ser nul.

cognom2

Camp alfanumèric de longitud 20.
Primer cognom del pacient. No pot ser nul, el tractaríem igual que el segon cognom del pacient.

professio

Camp alfanumèric de longitud 20.
Identifica la professió: metge, psicòleg, infermer, etc. No pot ser nul.

especialitat

Camp alfanumèric de longitud 20.
Especialitat del professional. Per exemple: cardiologia, traumatologia, cures infermeria, etc. No pot ser nul·la.

numCol_legiat

Camp numèric de longitud 5.
Número de col·legiat del professional. No pot ser nul.

6.- HORAVISITA

Guarda les hores de visita.

Clau primària formada per: **diaHora** i **id_professional**.

diaHora

Camp amb format data.

Dia i hora de la visita. No pot ser nul·la.

id_professional

Camp numèric de longitud 4.

Identifica el professional al que se li assigna la visita. No pot ser nul.

tempsMinim

Camp numèric de longitud 2.

Temps mínim de la visita en minuts. No pot ser nul.

numHistorial

Camp numèric de longitud 6.

Identifica el pacient a visitar. Pot ser nul.

7.- FITXA

Guarda les dades del contingut de la visita.

Clau primària formada per: **numHistorial** i **diaHora**.

numHistorial

Camp numèric de longitud 6.

Identifica el pacient que es visita. No pot ser nul.

diaHora

Camp amb format data.

Dia i hora de la visita. No pot ser nul.

id_professional

Camp numèric de longitud 4.

Nom del professional que té assignada l'hora de visita.

professionalVisita

Camp numèric de longitud 4.

Nom del professional que visita el pacient, no ha de coincidir necessàriament amb el professional que tenia assignada l'hora de visita (podria ser un substitut). No pot ser nul.

al_lergies

Camp alfanumèric de longitud 50.

Al·lèrgies del pacient, per defecte serà: 'No conegudes'.

Pot ser nul.

anotacions

Camp alfanumèric de longitud 255.

Anotacions del professional respecte a la visita.

No pot ser nul.

diagnostic

Camp alfanumèric de longitud 30.

Diagnòstic del pacient. No pot ser nul.

altaMedica

Camp alfanumèric de longitud 2.

Valors possibles: SI, NO. No pot ser nul.

Indica si el metge dona d'alta el pacient del seu procés assistencial. Per defecte serà 'NO', només si el professional és metge podria ser 'SI'.

8.- MEDICAMENT

Guarda tots els medicaments.

Clau primària formada per: **nomComercial**.

nomComercial

Camp alfanumèric de longitud 30.

Nom comercial del medicament. No pot ser nul.

nomGeneric

Camp alfanumèric de longitud 30.

Nom genèric del medicament. No pot ser nul.

grupFarmacologic

Camp alfanumèric de longitud 20.

Grup farmacològic al qual pertany el medicament. No pot ser nul.

9.- ASSIGNACIO PACIENT

Interrelació entre PACIENT i HORAVISITA.

Un pacient pot tenir assignada més d'una hora de visita.

Una hora de visita pot ser assignada només a un pacient.

Pot haver-hi pacients que no tinguin assignada una hora de visita.

Pot haver-hi hores de visita sense pacient assignat.

10.- ASSIGNACIO PROFESSIONAL

Interrelació entre PROFESSIONAL i HORAVISITA.

Un professional pot tenir assignades més d'una hora de visita.

Una hora de visita només pot tenir un professional assignat.

Tota hora de visita té un professional assignat.

Pot haver-hi professionals sense una hora de visita assignada.

11.- HISTORIAL

Interrelació entre PACIENT i FITXA MEDICA, correspon al historial mèdic, ja que aquest no és més que un pacient amb un conjunt de fitxes mèdiques.

Un pacient pot tenir més d'una fitxa mèdica.

Una fitxa mèdica només pot ser d'un pacient.

Pot haver-hi pacients sense fitxa mèdica.

Tota fitxa mèdica és d'un pacient.

12.- VISITA

Interrelació entre HORAVISITA i FITXA.

Una hora de visita pot tenir només una fitxa mèdica.

Una fitxa correspon a una única hora de visita.

Pot haver-hi hores de visita sense fitxa associada (ex. Durant el temps que passa entre que s'assigna l'hora de visita i quan es produeix la visita).

No hi pot haver una visita sense hora de visita.

13.- FA VISITA

Interrelació entre PROFESSIONAL i FITXA.

Un professional pot fer cap, una o més fitxes mèdiques.

Una fitxa ha de ser feta per un sol professional.

No hi ha cap fitxa que no hagi estat feta per un professional.

Pot haver-hi professionals que no hagin fet una visita.

14.- RECEPTA

Interrelació entre MEDICAMENT i FITXA.

En una visita mèdica es poden receptar cap, un o més medicaments.

Un medicament pot ser receptat en cap, una o més visites mèdiques.

No tots els medicaments són receptats.

No en totes les fitxes mèdiques es recreen medicaments.

15.- LOG_TAUULA

Guarda les dades relacionades amb els accessos a la base de dades.

tipus_operacio

Camp alfanumèric de longitud 30.

Identifica l'operació que es fa sobre la base de dades.

idUsuari

Camp numèric.

Identifica a l'usuari que accedeix a la base de dades.

dataHora

Camp amb format data i hora.

Identifica el moment de l'accés a la base de dades.

paramEntrada

Camp alfanumèric de longitud 512.

Paràmetres d'entrada del procediment.

paramSortida

Camp alfanumèric de longitud 255.

Paràmetres de sortida del procediment.

16.- PARAMETRES

Guarda els paràmetres del sistema: temps mínim de la visita, horari consultori, etc.

Clau primària formada per: **nom**.

nom

Camp alfanumèric de longitud 20.

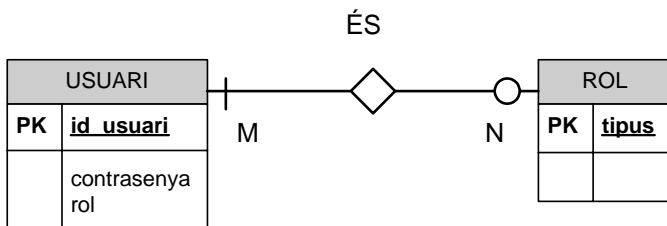
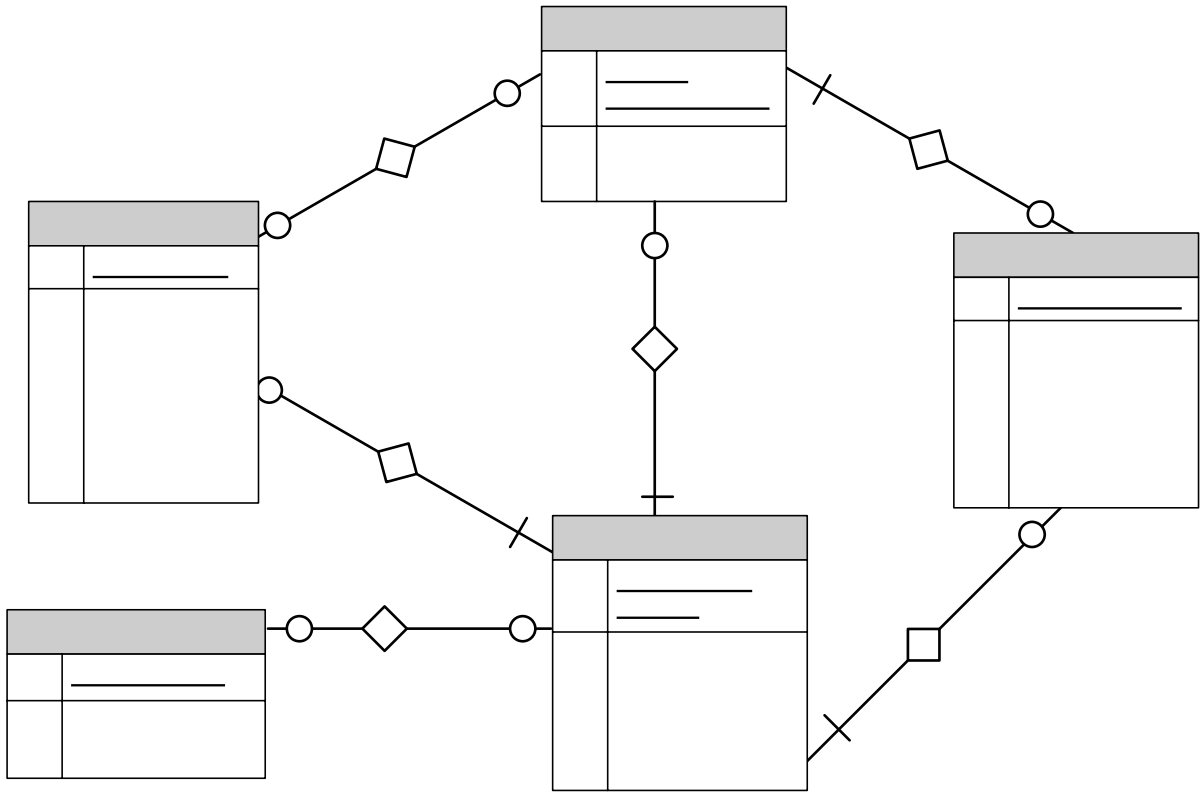
Identifica el paràmetre.

valor

Camp alfanumèric de longitud 20.

El valor del paràmetre. No pot ser nul.

4.1.2.- DIAGRAMA ER



LOG_TAUOLA	
PK	id_usuari
PK	dataHora
	<i>tipus_operacio</i>
	<i>paramEntrada</i>
	<i>paramSortida</i>

	—

PACIENT

PK numHistorial

dni

ASS
P

4.2.- DISSENY LÒGIC

En aquesta fase partim del resultat anterior, el disseny conceptual. Les entitats originaran relacions i les interrelacions poden donar lloc a claus foranes o una nova relació.

PACIENT (numHistorial, dni, nom, cognom1, cognom2, dataNaixement, adreça, telefon)

PROFESSIONAL (id_professional, nom, cognom1, cognom2, professio, especialitat, numCol_legiat)

En la relació visita, entre HORAVISITA i FITXA, s'ha optat per posar la clau forana a Fitxa. La clau forana ha de fer referència a ocurrences úniques, i tot i que un pacient no pot ser visitat alhora per dos professionals, si que pot passar que hi hagin dues hores de visita amb *numHistorial* nul. Per tant s'ha hagut d'afegir *id_professional* a la relació fitxa per crear la clau forana (*dia_hora,id_professional,numHistorial*).

HORAVISITA (diaHora, id_professional, tempsMinim, numHistorial)
 on {id_professional} és clau forana cap a PROFESSIONAL
 on {numHistorial} és clau forana cap a PACIENT

FITXA (numHistorial, diaHora, id_professional, professionalVisita, al_lergies, anotacions, diagnostic, altaMedica)
 on {numHistorial} és clau forana cap a PACIENT
 on {id_professional} és clau forana cap a PROFESSIONAL
 on {professionalVisita} és clau forana cap a PROFESSIONAL
 on {diaHora,id_professional,numHistorial} és clau forana cap a HORAVISITA.

MEDICAMENT (nomComercial, nomGeneric, grupFarmacologic)

RECEPTA (nomComercial, diaHora, numHistorial)
 on {nomComercial} és clau forana cap a MEDICAMENT
 on {numHistorial, diaHora} és clau forana cap a FITXA
 on {numHistorial} és clau forana cap a PACIENT

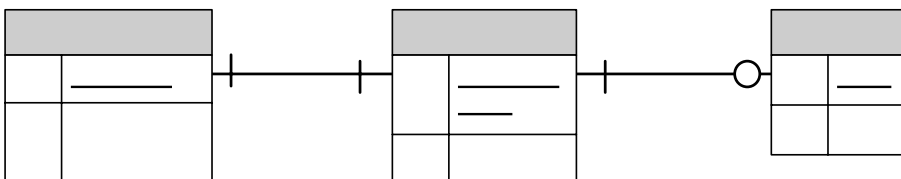
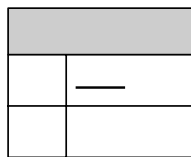
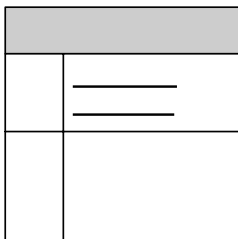
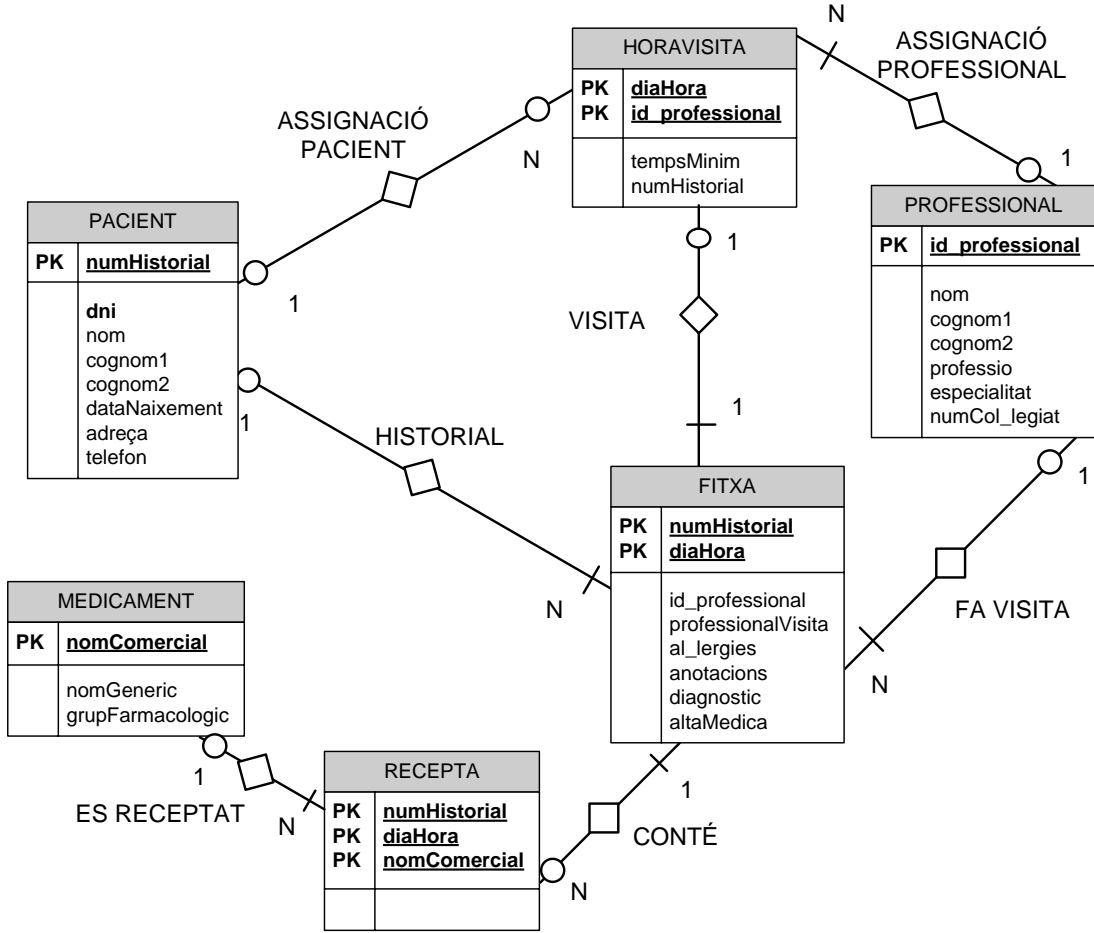
ROL (tipus)

USUARI (id_usuari, contrasenya, rol)
 ON [rol] és clau forana cap a ROL

PARAMETRES (nom, valor)

LOG_TAULA (id_usuari, dataHora, tipus_operacio, paramEntrada, paramSortida)

4.2.1.- DIAGRAMA ER FINAL



4.2.2.- RESTRICCIONS D'INTEGRITAT

Tractament de les claus foranes

- Si s'esborra un pacient i existeix una hora de visita, aquesta prendrà el valor nul per a l'atribut *numHistorial*.
- No es pot esborrar un professional si hi ha una hora de visita que el faci referència.
- No es pot esborrar un pacient si hi ha una fitxa que el faci referència.
- No es pot esborrar un professional si hi ha una fitxa que el faci referència.
- No es pot esborrar una hora de visita si hi ha una fitxa que la fa referència.
- No es pot esborrar un medicament si hi ha una recepta que el fa referència.
- Si s'esborra una fitxa, s'esborraran les receptes associades.

Altres restriccions d'integritat

- L'atribut *altaMedica* de la relació FITXA només pot ser 'SI' si el *professionalVisita* és metge.
- En una fitxa els professionals *id_professional* i *professionalVisita* han de tenir la mateixa professió.
- El professional que fa la recepta ha de ser metge, i coincidir amb el *professionalVisita*.
- No pot haver-hi dos professionals amb la mateixa professió i mateix número de col·legiat.

4.2.3.- REGLES DE NEGOCI

Restriccions que han de complir les operacions sobre les dades:

- D'un pacient només es poden modificar l'adreça i el telèfon, o afegir DNI si no en tenia.
- Un pacient no es pot esborrar si hi ha una fitxa mèdica que el fa referència. Per tant, l'esborrat d'un pacient implica esborrar primer les seves fitxes i receptes.
- Una fitxa no es pot esborrar si no es fa en conjunt (tot el historial) i han passat com a mínim 5 anys des de la data d'alta mèdica de l'últim procés assistencial.
- No es pot modificar una fitxa.
- D'un professional només es pot modificar l'especialitat.
- Un medicament no es pot modificar.
- Una recepta no es pot modificar.

4.3.- DISSENY FÍSIC

En aquesta fase es tradueix l'esquema lògic en un que es pugui implementar en Oracle, la qual cosa implica conèixer primer la funcionalitat que aquest ofereix.

Primer es dissenyen les relacions base, amb els atributs corresponents (especificant el tipus de dades, longitud, restriccions de domini, valor per defecte, si admeten nuls, etc.), claus primàries i foranes, i les regles d'integritat de les claus foranes.

4.3.1. RELACIONS

1.- Relació PACIENT

```
-- Taula PACIENT
create table PACIENT(
  numHistorial number(6) constraint PACIENT_PK primary key,
  dni number(8) constraint PACIENT_DNI_UQ unique,
  nom varchar2(20) constraint PACIENT_NOM_NN not null,
  cognom1 varchar2(20) constraint PACIENT_COGNOM1_NN not null,
  cognom2 varchar2(20) constraint PACIENT_COGNOM2_NN not null,
  dataNaixement date constraint PACIENT_DATANAIXEMENT_NN not null,
  adreça varchar2(50) constraint PACIENT_ADREÇA_NN not null,
  telefon number(9) constraint PACIENT_TELEFON_CK
  check (TELEFON between 100000000 and 999999999),
  constraint PACIENT_COG_NOM_ADR_DNI_UQ unique(cognom1, cognom2, nom, adreça, dni)
);
```

Per a les dades alfanumèriques s'ha optat per *varchar2*, d'aquesta manera si la dada ocupa menys espai de l'assignat per a l'atribut, Oracle només assignarà l'espai necessari.

Donat que *dni* pot ser nul, s'ha escollit com a clau alternativa (*cognom1*, *cognom2*, *nom*, *adreça*, *dni*).

2. Seqüència S_NUMHISTORIAL

Oracle no té un tipus de dades numèriques incrementables automàticament, però podem crear un objecte seqüència (SEQUENCE) com a generador de números per a la clau primària *numHistorial*:

```
-- Seqüència per a numHistorial
create sequence s_Numhistorial
  start with 1
  increment by 1
  maxvalue 999999
  nocycle
  nocache;
```

La numeració comença per 1 i s'incrementa d'un en un amb un màxim de 999.999. Quan arriba al final, no tornarà a començar de nou per l'1 (*nocycle*) ja que encara podríem tenir emmagatzemat el primer historial. Amb *nocache* evitem que es generi automàticament més d'un número (els números generats es guarden en memòria i així el temps de resposta és menor), d'aquesta manera evitem perdre els números que no s'assignin en una sessió.

3. Relació PROFESSIONAL

```
-- Taula PROFESSIONAL
create table PROFESSIONAL(
  id_professional number(4) constraint PROFESSIONAL_PK primary key,
  nom varchar2(20) constraint PROFESSIONAL_NOM_NN not null,
  cognom1 varchar2(20) constraint PROFESSIONAL_COGNOM1_NN not null,
  cognom2 varchar2(20) constraint PROFESSIONAL_COGNOM2_NN not null,
  professio varchar2(20) constraint PROFESSIONAL_PROFESSIO_NN not null,
  especialitat varchar2(20) constraint PROFESSIONAL_ESPECIALITAT_NN not null,
  numCol_legiat number(5) constraint PROFESSIONAL_NUMCOL_LEGGIAT_NN not null,

  constraint PROFESSIONAL_ESPEC_NUMCOL_UQ unique (professio,numCol_legiat)
);
```

Amb la restricció PROFESSIONAL_ESPEC_NUMCOL_UQ s'assegura que no hi haurà dos professionals amb la mateixa professió i número de col·legiat.

4. Seqüència S_ID_PROFESSIONAL

També es farà us d'una seqüència per generar la clau primària *id_professional*:

```
-- Seqüència per a id_professional
create sequence s_id_professional
  start with 1
  increment by 1
  maxvalue 9999
  nocycle
  nocache;
```

5. Relació HORAVISITA

En aquesta relació existeixen claus foranes on s'han d'aplicar restriccions d'integritat:

- ON DELETE SET NULL: si s'elimina l'ocurrència a la que fa referència la clau forana, aquesta prendrà el valor nul. Per exemple, si s'esborra un pacient l'hora de visita corresponent prendrà el valor nul per a l'atribut *numHistorial*.
- Si no s'especifica res, Oracle evita que l'ocurrència a la que fa referència la clau forana es pugui eliminar. Per exemple, no es podrà esborrar un professional si existeix alguna ocurrència en HORAVISITA d'aquest professional.
- ON DELETE CASCADE: si s'elimina l'ocurrència a la que fa referència la clau forana s'eliminarà també la ocurrència on es troba aquesta. No es fa servir en aquesta relació.

```
-- Taula HORAVISITA
create table HORAVISITA(
  diaHora timestamp(0),
  id_professional number(4) constraint HORAVISITA_ID_PROFESSIONAL_FK
    references PROFESSIONAL(ID_PROFESSIONAL),
  tempsMinim number(2) constraint HORAVISITA_TEMPSPMINIM_NN not null,
  numHistorial number(6) constraint HORAVISITA_NUMHISTORIAL_FK
    references PACIENT(NUMHISTORIAL)
    on delete set null,
  constraint HORAVISITA_PK primary key (diaHora,id_professional),
  constraint HORAVISITA_DH_PROF_NHIST_UQ
    unique (diaHora,id_professional,numHistorial)
);
```

6. Relació FITXA

En aquesta relació també hi ha claus foranes, per defecte no es podrà esborrar un pacient si hi ha una fitxa que el fa referència. Igualment per als professionals i les hores de visita.

Amb la restricció FITXA_ALTAMEDICA_CK s'assegura que el valor de *altaMedica* és 'SI' o 'NO'.

```
-- Taula FITXA
create table FITXA(
  numHistorial number(6) constraint FITXA_NUMHISTORIAL_FK
                                references PACIENT(NUMHISTORIAL),
  diaHora timestamp(0),
  id_professional number(4) constraint FITXA_ID_PROFESSIONAL_NN not null
                              constraint FITXA_ID_PROFESSIONAL_FK
                              references PROFESSIONAL(ID_PROFESSIONAL),
  professionalVisita number(4) constraint FITXA_PROFESSIONALVISITA_NN not null
                              constraint FITXA_PROFESSIONALVISITA_FK
                              references PROFESSIONAL(ID_PROFESSIONAL),
  al_lergies varchar2(50) default ('No conegudes')
                              constraint FITXA_AL_LERGIAS_NN not null,
  anotacions varchar2(255) constraint FITXA_ANOTACIONES_NN not null,
  diagnostic varchar2(30) constraint FITXA_DIAGNOSTIC_NN not null,
  altaMedica char(2) default ('NO') constraint FITXA_ALTAMEDICA_NN not null
                                   constraint FITXA_ALTAMEDICA_CK
                                   check (altaMedica in ('SI','NO')),
  constraint FITXA_PK primary key (numHistorial, diaHora),
  constraint FITXA_DH_PROF_NHIST_FK
            foreign key (diaHora, id_professional, numHistorial)
            references HORAVISITA(DIAHORA, ID_PROFESSIONAL, NUMHISTORIAL)
);
```

7. Relació MEDICAMENT

```
-- Taula MEDICAMENT
create table MEDICAMENT(
  nomComercial varchar2(30) constraint MEDICAMENT_NOMCOMERCIAL_PK primary key,
  nomGeneric varchar2(30) constraint MEDICAMENT_NOMGENERIC_NN not null,
  grupFarmacologic varchar2(20) constraint MEDICAMENT_GRUPFARMACOLOGIC_NN not null
);
```

8. Relació RECEPТА

Aquí també hi ha claus foranes:

- Per defecte, no es pot esborrar un pacient, un medicament o un professional si hi ha una recepta que els fa referència.
- Si s'esborra una fitxa s'eliminaran les receptes associades a aquesta (*on delete cascade*).

```
-- Taula RECEPТА
create table RECEPТА(
  numHistorial number(6) constraint RECEPТА_NUMHISTORIAL_FK
                                references PACIENT(NUMHISTORIAL),
  diaHora timestamp(0),
  nomComercial varchar2(30) constraint RECEPТА_NOMCOMERCIAL_FK
                              references MEDICAMENT(NOMCOMERCIAL),
  id_professional number(4) constraint RECEPТА_ID_PROFESSIONAL_NN not null,
  constraint RECEPТА_PK primary key (numHistorial, diaHora, nomComercial),
  constraint RECEPТА_NUMHIST_DIAHORA_FK foreign key (numHistorial, diaHora)
            references FITXA(NUMHISTORIAL, DIAHORA)
            on delete cascade
);
```

9. Relació PARAMETRES

```
-- Taula PARAMETRES
create table PARAMETRES(
  nom varchar2(20) constraint PARAMETRES_NOM_PK primary key,
  valor varchar2(20) constraint PARAMETRES_VALOR_NN not null
);

insert into parametres values('Hora inici','09:00');
insert into parametres values('Hora fi','13:00');
insert into parametres values('Temps mínim','30');
```

En aquesta relació tenim els paràmetres del consultori: el temps mínim de la visita és de 30 minuts, i l'horari és de 9 h a 13 h.

10. Relació LOG_TAULA

```
-- Taula LOG_TAULA
create table LOG_TAULA(
  tipus_operacio varchar2(30) not null,
  id_usuari number not null,
  dataHora timestamp not null,
  paramEntrada varchar2(512) not null,
  paramSortida varchar2(255) not null,
);
```

4.3.2. DISPARADORS

Amb els disparadors es pot:

- garantir les restriccions d'integritat que no poden ser definides a nivell d'esquema,
- garantir el compliment de les regles de negoci,
- mantenir la informació derivada,
- evitar l'execució de transaccions invàlides.

1. Disparador INS_PACIENT

```
-- Disparador INS_PACIENT
create or replace trigger ins_pacient
  before insert on pacient
  for each row
  declare
  begin
    select s_numhistorial.nextval into :new.numHistorial from dual;
  end ins_pacient;
/
```

S'executa sempre abans d'inserir un pacient a la taula PACIENT, assigna a *numHistorial* el número generat per la seqüència *s_numHistorial*.

2. Disparador INS_PROFESSIONAL

```
-- Disparador INS_PROFESSIONAL
create or replace trigger ins_professional
  before insert
  on professional
  for each row
  begin
    select s_id_professional.nextval into :new.id_professional from dual;
  end ins_professional;
/
```

S'executa sempre abans d'inserir un professional a la taula PROFESSIONAL, assigna a *id_professional* el número generat per la seqüència *s_id_professional*.

3. Disparador INS_FITXA

S'executa sempre abans d'inserir una fitxa a la taula FITXA, comprova en el cas que els professionals siguin diferents, que aquests tinguin la mateixa professió. A més comprova si l'atribut *altaMedica* és 'SI', que el professional que fa la visita sigui metge. En tots dos casos, si no es compleixen les restriccions, s'atura l'execució i s'informa del motiu.

```

-- Disparador INS_FITXA
create or replace trigger ins_fitxa
before insert
on FITXA
for each row
declare
    professio1 varchar2(20);
    professio2 varchar2(20);
    vtrace number := 1;
begin
    select professio into professio1 from professional
        where (id_professional = :new.id_professional);
    vtrace := 2;
    select professio into professio2 from PROFESSIONAL
        where (id_professional = :new.professionalVisita);

    if (:new.id_professional <> :new.professionalVisita) then
        if (professio1 <> professio2) then
            raise_application_error
                (-20010, 'ERROR: El professional suplent ha de tenir la mateixa
professió que el titular. ');
            end if;
        end if;

        if ((:new.altaMedica = 'SI') and (professio2 <> 'Metge')) then
            raise_application_error
                (-20011, 'ERROR: Una alta mèdica només la pot fer un metge. ');
            end if;

        exception
            when no_data_found then
                if vtrace = 1 then
                    raise_application_error
                        (-20012, 'ERROR: No existeix el professional titular. ');
                else
                    raise_application_error
                        (-20013, 'ERROR: No existeix el professional suplent. ');
                end if;
            end ins_fitxa;
/

```

4. Disparador INS_RECEPTA

S'executa sempre abans d'inserir una recepta a la taula RECEPTA, comprova si el professional és metge i coincideix amb el que fa la visita. Si no es compleixen les restriccions, s'atura l'execució i s'informa del motiu.

```

-- Disparador INS_RECEPTA
create or replace trigger ins_recepta
before insert
on RECEPTA
for each row
declare
    vtrace number := 1;
    xprof varchar2(20);
    xsuplent number(4);
begin
    select professio into xprof from professional
        where (id_professional = :new.id_professional) ;
    if (xprof <> 'Metge') then
        raise_application_error (-20014, 'ERROR: Només un metge pot fer receptes. ');
    end if;

    vtrace := 2;
    select professionalVisita into xsuplent from fitxa where
        ((numHistorial = :new.numHistorial) and (diaHora = :new.diaHora));

    if (:new.id_professional <> xsuplent) then
        raise_application_error
            (-20015, 'ERROR: El metge que visita el pacient és el que ha de fer la
recepta. ');
    end if;
end if;

```

```

end if;

exception
  when no_data_found then
    if vtrace = 1 then
      raise_application_error (-20016,'ERROR: Metge no trobat.');
```

```

    else
      raise_application_error
        (-20017,'ERROR: No existeix una fitxa d'aquest pacient i dia i hora.');
```

```

    end if;
end ins_recepta;
/
```

5. Disparador UPD_PACIENT

```

-- Disparador UPD_PACIENT
create or replace trigger upd_pacient
  before update of numHistorial,nom,cognom1,cognom2,dataNaixement
  on PACIENT
  begin
    raise_application_error
      (-20018,'ERROR: Només es pot modificar adreça i/o telèfon, o afegir DNI.');
```

```

  end upd_pacient;
/
```

S'executa sempre abans de modificar els atributs: *numHistorial*, *nom*, *cognom1*, *cognom2* i *dataNaixement* de la taula PACIENT. S'atura l'execució i s'informa del motiu.

6. Disparador UPD_PACIENT_DNI

```

-- Disparador UPD_PACIENT_DNI
create or replace trigger upd_pacient_dni
  after update of dni
  on PACIENT
  for each row
  when ((old.dni is not null) and (new.dni != old.dni))
  begin
    raise_application_error
      (-20019,'ERROR: Només es pot afegir DNI si no en tenia.');
```

```

  end upd_pacient_dni;
/
```

S'executa sempre després de modificar l'atribut *dni* de la taula PACIENT, si el pacient ja tenia un DNI i el nou és diferent atura l'execució i informa del motiu.

7. Disparador UPD_PROFESSIONAL

```

-- Disparador UPD_PROFESSIONAL
create or replace trigger upd_professional
  before update of id_professional,nom,cognom1,cognom2,professio,numCol_legiat
  on PROFESSIONAL
  begin
    raise_application_error
      (-20020,'ERROR: Només es pot modificar l''especialitat
professional.');
```

```

  end upd_professional;
/
```

S'executa sempre abans de modificar els atributs: *id_professional*, *nom*, *cognom1*, *cognom2*, *professio*, *numCol_legiat* de la taula PROFESSIONAL. S'atura l'execució i s'informa del motiu.

8. Disparador UPD_HORAVISITA

```
-- Disparador UPD_HORAVISITA
create or replace trigger upd_horaVisita
  before update of diaHora,id_professional,tempsMinim
  on HORAVISITA
  begin
    raise_application_error
      (-20021,'ERROR: Només es pot modificar el número d''historial.');
```

S'executa sempre abans de modificar els atributs: *diaHora*, *id_professional* i *tempsMinim* de la taula HORAVISITA. S'atura l'execució i s'informa del motiu.

9. Disparador UPD_FITXA

```
-- Disparador UPD_FITXA
create or replace trigger upd_fitxa
  before update
  on FITXA
  begin
    raise_application_error
      (-20024,'ERROR: No es pot modificar el contingut d''una fitxa');
```

S'executa sempre abans de modificar qualsevol atribut de la taula FITXA. S'atura l'execució i s'informa del motiu.

10. Disparador UPD_MEDICAMENT

```
-- Disparador UPD_MEDICAMENT
create or replace trigger upd_medicament
  before update
  on MEDICAMENT
  begin
    raise_application_error (-20025,'ERROR: No es pot modificar un medicament.');
```

S'executa sempre abans de modificar qualsevol atribut de la taula MEDICAMENT. S'atura l'execució i s'informa del motiu.

11. Disparador UPD_RECEPTA

```
-- Disparador UPD_RECEPTA
create or replace trigger upd_recepta
  before update
  on RECEPTA
  begin
    raise_application_error (-20026,'ERROR: No es pot modificar una recepta.');
```

S'executa sempre abans de modificar qualsevol atribut de la taula RECEPTA. S'atura l'execució i s'informa del motiu.

12. Paquet GESTIOCONSULTORI

```
-- Paquet GESTIOCONSULTORI
create or replace package GestioConsultori is

    v_periode5anys interval year to month := numtoyminterval(5,'YEAR');
    v_semaforFitxa boolean := false;

end GestioConsultori;
/
```

En aquest paquet tenim dues variables globals (poden ser consultades per tots els disparadors i procediments).

13. Disparador DEL_RECEPTA

```
-- Disparador DEL_RECEPTA
create or replace trigger del_recepta
    after delete --AFTER, així si s'intenta esborrar una fitxa, donarà l'error de fitxa.
    on RECEPTA

    begin
        if GestioConsultori.v_semaforFitxa = false then
            raise_application_error
                (-20027,'ERROR: No es pot esborrar una recepta si no s'esborra també la
fitxa.');
```

Comprova el valor de la variable `v_semaforFitxa` del paquet `GESTIOCONSULTORI`, si és `false`, s'atura l'execució i s'informa del motiu.

14. Disparador DEL_FITXA

Comprova el valor de la variable `v_semaforFitxa` del paquet `GESTIOCONSULTORI`, si és `false`, s'atura l'execució i s'informa del motiu.

```
-- Disparador DEL_FITXA
create or replace trigger del_fitxa
    before delete
    on FITXA

    begin
        if GestioConsultori.v_semaforFitxa = false then
            raise_application_error (-20028,'ERROR: No es pot esborrar la fitxa.');
```

4.3.3. PROCEDIMENTS EMMAGATZEMATS

Degut a l'extensió del codi dels procediments emmagatzemats, aquest es troba a l'arxiu: **mpradera_annex1_codi_procediments.doc**.

Tots els procediments:

- Insereixen a LOG_TAULA el nom del procediment, l'usuari, el dia i l'hora, els paràmetres d'entrada i els de sortida.
- Tenen un paràmetre de sortida pRSP que conté:
 - **'OK' + acció realitzada**, si l'execució finalitza amb èxit.
 - **'ERROR' + nom del procediment + descripció de l'error**, si es produeix algun error definit. Si és el cas, es capturen els errors que llencen els disparadors, d'aquesta manera evitem fer comprovacions addicionals i simplifiquem el codi dins dels procediments.
 - **'ERROR' + traça + codi error SQL + missatge error SQL**, si es produeix un error inesperat durant l'execució.

En els casos dels procediments: INSERIR_FITXA, MODIFICAR_FITXA, ESBORRAR_FITXES i ESBORRAR_PACIENT, el paràmetre *pRSP* és d'entrada i de sortida, d'aquesta manera comuniquem a través de *pRSP* el procediments que són cridats per un altre.

1. Procediment INSERIR_PACIENT

PRECONDICIONS:

- els paràmetres pnom, pcognom1, pcognom2, pdataNaixement i padreça no poden ser nuls,
- el pacient no existeix a la base de dades.

POSTCONDICIONS:

- afegeix un nou pacient a la taula PACIENT,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Pacient inserit.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

2. Procediment MODIFICAR_PACIENT

PRECONDICIONS:

- els paràmetres pnumHistorial i padreça i ptelefon no poden ser nuls,
- el pacient existeix a la taula PACIENT,
- si l'atribut *dni* anterior del pacient és diferent de nul, el nou ha de ser el mateix.

POSTCONDICIONS:

- afegeix un dni al pacient i/o modifica l'adreça i/o modifica el telèfon,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Pacient modificat.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

3. Procediment INSERIR_PROFESIONAL

PRECONDICIONS:

- els paràmetres pnom, pcognom1, pcognom2, pprofessio, pespecialitat i pnumCol_legiat no poden ser nuls,
- el professional no existeix a la base de dades.

POSTCONDICIONS:

- afegeix un nou professional a la taula PROFESSIONAL,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Professional inserit.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4. Procediment MODIF_ESPECIALITATPROFESIONAL

PRECONDICIONS:

- els paràmetres pidProfessional i pespecialitat no poden ser nuls,
- el professional existeix a la taula PROFESSIONAL.

POSTCONDICIONS:

- modifica l'especialitat professional,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Especialitat professional modificada.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

5. Procediment CREAR_HORES_VISITA

PRECONDICIONS:

- els paràmetres pdia i pprofessional no poden ser nuls,
- el professional existeix a la taula PROFESSIONAL,
- no existeix a la taula HORAVISITA cap de les hores de visita a crear.

POSTCONDICIONS:

- crea les hores de visita d'un professional de tot un dia, segons el temps mínim i l'horari de la taula PARAMETRES (numHistorial es nul fins que s'assigni l'hora de visita al pacient),

- el paràmetre de sortida pRSP tindrà el valor: 'OK: Creades hores de visita.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

6. Procediment ASSIGNAR_HORA_VISITA

PRECONDICIONS:

- els paràmetres pnumHistorial, pidProfessional i pdiaHora no poden ser nuls,
- existeix l'hora de visita a la taula HORAVISITA,
- existeix el pacient a la taula PACIENT,
- l'hora de visita no està assignada a cap pacient.

POSTCONDICIONS:

- assigna l'hora de visita al pacient,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Assignada hora de visita.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

7. Procediment DESASSIGNAR_HORA_VISITA

PRECONDICIONS:

- els paràmetres pnumHistorial, pidProfessional i pdiaHora no poden ser nuls,
- existeix l'hora de visita a la taula HORAVISITA,
- el pacient té assignada l'hora de visita,
- no hi ha una fitxa associada a l'hora de visita.

POSTCONDICIONS:

- posa a nul el numHistorial de l'hora de visita,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Desassignada hora de visita.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

8. Procediment INSERIR_FITXA

PRECONDICIONS:

- els paràmetres pnumHistorial, pdiaHora, pidProfessional, pProfessionalVisita, pAnotacions, pDiagnostic i pAltaMedica no poden ser nuls,
- el paràmetre pRSP té un valor diferent de nul si aquest procediment és cridat per un altre,
- no existeix la fitxa a la taula FITXA,
- existeix una hora de visita (del professional i amb el pacient assignat) per a la fitxa a la taula HORAVISITA,
- existeix el professional suplent a la taula PROFESSIONAL,
- si el professional titular i el suplent són diferents, tenen la mateixa professió,
- si l'atribut altaMedica és 'SI', el professional és 'Metge'.

POSTCONDICIONS:

- si no s'introdueix un valor per a pAl_lergies, prendrà el valor: 'No conegudes',
- afegeix una fitxa a la taula FITXA,

- si aquest procediment no es crida per un altre (pRSP era nul) llavors el paràmetre de sortida pRSP tindrà el valor: 'OK: Fitxa inserida.' si l'execució finalitza amb èxit, altrament indica l'error produït. A més, insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida,
- si és crida per un altre procediment, aquest retornarà a través del paràmetre pRSP el valor que tenia si l'execució finalitza amb èxit, en cas contrari pRSP continuarà l'error produït perquè sigui tractat pel procediment que l'ha crida.

9. Procediment MODIFICAR_FITXA

PRECONDICIONS:

- els paràmetres pnumHistorial, pdiaHora, pidProfessional, pProfessionalVisita, pAnotacions, pDiagnostic i paltaMedica no poden ser nuls,
- existeix la fitxa a modificar a la taula FITXA.

POSTCONDICIONS:

- si no s'introdueix un valor per a pAl_lergies, prendrà el valor: 'No conegudes',
- crea una hora de visita per a la nova fitxa a la taula HORAVISITA,
- afegeix una fitxa (mitjançant el procediment INSERIR_FITXA) a la taula FITXA amb la data i l'hora del sistema,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Fitxa modificada.' si l'execució finalitza amb èxit, altrament indica l'error produït (si l'error es produeix al procediment inserir_fitxa serà del tipus: 'ERROR Modificar Fitxa + ERROR Inserir Fitxa ...'),
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

10. Procediment INSERIR_MEDICAMENT

PRECONDICIONS:

- els paràmetres pnomComercial, pnomGeneric i pgrupFarmacologic no poden ser nuls,
- el medicament no existeix a la taula MEDICAMENT.

POSTCONDICIONS:

- afegeix un nou medicament a la taula MEDICAMENT,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Medicament inserit.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

11. Procediment INSERIR_RECEPTA

PRECONDICIONS:

- els paràmetres pnumHistorial, pdiaHora, pnomComercial i pid_professional no poden ser nuls,
- la recepta no existeix a la taula RECEPTA,
- el medicament receptat existeix a la taula MEDICAMENT,
- existeix la fitxa on associar la recepta a la taula FITXA,
- el professional existeix a la taula PROFESSIONAL, és 'Metge' i és el que ha fet la visita.

POSTCONDICIONS:

- afegeix una nova recepta a la taula RECEPTA,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Recepta inserida.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

12. Procediment ESBORRAR_PROFESSIONAL**PRECONDICIONS:**

- el paràmetre pid_professional no pot ser nul,
- el professional existeix a la taula PROFESSIONAL,
- no existeix cap hora de visita del professional.

POSTCONDICIONS:

- esborra el professional de la taula PROFESSIONAL,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Professional esborrat.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

13. Procediment ESBORRAR_MEDICAMENT**PRECONDICIONS:**

- el paràmetre pnomComercial no pot ser nul,
- el medicament existeix a la taula MEDICAMENT,
- no existeix cap recepta d'aquest medicament.

POSTCONDICIONS:

- esborra el medicament de la taula MEDICAMENT,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Medicament esborrat.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

14. Procediment ESBORRAR_HORAVISITA**PRECONDICIONS:**

- els paràmetres pdiaHora i pid_professional no poden ser nuls,
- l'hora de visita existeix a la taula HORAVISITA,
- no existeix una fitxa associada a l'hora de visita,

POSTCONDICIONS:

- esborra l'hora de visita de la taula HORAVISITA,
- el paràmetre de sortida pRSP tindrà el valor: 'OK: Hora de Visita esborrada.' si l'execució finalitza amb èxit, altrament indica l'error produït,
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

15. Procediment ESBORRAR_FITXES

PRECONDICIONS:

- el paràmetre *pnumHistorial* no pot ser nul,
- el paràmetre *pRSP* té un valor diferent de nul si aquest procediment és cridat per un altre,
- si no es cridat per un altre procediment, ha d'existir alguna fitxa del pacient a la taula FITXA,
- la fitxa més recent del pacient ha de ser com a mínim de fa 5 anys i el valor de *altaMedica* ha de ser 'SI'.

POSTCONDICIONS:

- esborra les fitxes del pacient de la taula FITXA,
- esborra les receptes associades a les fitxes esborrades de la taula RECEPTEA,
- si aquest procediment no es cridat per un altre (*pRSP* era nul) llavors el paràmetre de sortida *pRSP* tindrà el valor: 'OK: Fitxes esborrades.' si l'execució finalitza amb èxit, altrament indica l'error produït. A més, insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida,
- si és cridat per un altre procediment, aquest retornarà a través del paràmetre *pRSP*:
 - el valor 'OK' si s'han esborrat les fitxes,
 - el valor 'No fitxes' si el pacient no tenia cap fitxa,
 - el valor de l'error, si es produeix un error, perquè sigui tractat pel procediment que l'ha cridat.

16. Procediment ESBORRAR_PACIENT

PRECONDICIONS:

- el paràmetre *pnumHistorial* no pot ser nul,
- existeix el pacient a la taula PACIENT,
- si el pacient té alguna fitxa, la més recent ha de ser com a mínim de fa 5 anys i el valor de *altaMedica* ha de ser 'SI'.

POSTCONDICIONS:

- esborra les fitxes del pacient, si en té alguna, de la taula FITXA,
- s'esborren les receptes associades a les fitxes esborrades,
- esborra el pacient de la taula PACIENT,
- si el pacient tenia alguna hora de visita assignada, aquesta prendrà el valor nul per a *numHistorial*,
- el paràmetre de sortida *pRSP* tindrà el valor: 'OK: Pacient esborrat.' si l'execució finalitza amb èxit, altrament indica l'error produït (si l'error es produeix al procediment *esborrar_fitxes*, serà del tipus: 'ERROR Esborrar Pacient + ERROR Esborrar Fitxes ...'),
- insereix a la taula LOG_TAULA el nom del procediment, l'usuari, l'hora, els paràmetres d'entrada i el de sortida.

4.3.4. ÍNDEXS SECUNDARIS

Els índexs són estructures que es dissenyen per augmentar el rendiment, permeten un accés més ràpid a les dades d'una taula.

Abans d'assignar o anul·lar hores de visita es necessita conèixer el *numHistorial* del pacient, la qual cosa implica fer primer una consulta a la taula PACIENT. Els atributs més freqüents sobre els que es farà la consulta són: *cognom1*, *cognom2*.

Això ens pot fer pensar en crear un índex secundari per a aquets atributs. Oracle però, crea automàticament els índexs que afecten a les claus primàries i secundàries (*unique*). Ja hi ha doncs, un índex per (*cognom1*, *cognom2*, *nom*, *adreça*, *dni*).

Respecte als professionals, també pot ser freqüent realitzar consultes per conèixer l'*id_professional*, però no es aconsellable crear índexs sobre relacions petites com pot ser aquesta.

Per tant, després d'analitzar les consultes i transaccions que s'executaran sobre la base de dades, s'ha optat per no crear, de moment, cap índex secundari.

4.3.5. USUARIS

Per simplificar la gestió d'usuaris de la base de dades, es crearan quatre rols amb diferents privilegis, corresponents a les funcionalitats especificades en els requisits d'usuari de l'etapa d'anàlisi. A cada usuari creat se li assignarà com a mínim un rol.

1. Rol ADMINISTRADOR_BD

```
-- ROL ADMINISTRADOR_BD
-- Create the role
create role Administrador_BD;
-- Grant/Revoke role privileges
grant dba to Administrador_BD;
```

2. Rol ADMINISTRATIU

```
-- ROL ADMINISTRATIU
-- Create the role
create role ADMINISTRATIU;
-- Grant/Revoke object privileges
grant select on PACIENT to ADMINISTRATIU;
grant select on HORAVISITA to ADMINISTRATIU;
grant execute on ASSIGNAR_HORA_VISITA to ADMINISTRATIU;
grant execute on DESASSIGNAR_HORA_VISITA to ADMINISTRATIU;
grant execute on INSERIR_PACIENT to ADMINISTRATIU;
grant execute on MODIFICAR_PACIENT to ADMINISTRATIU;
-- Grant/Revoke system privileges
grant create session to ADMINISTRATIU;
```


3. Rol PROFESSIONAL_SANITARI

```
-- ROL PROFESSIONAL_SANITARI

-- Create the role
create role PROFESSIONAL_SANITARI;

-- Grant/Revoke object privileges
grant select on PACIENT to PROFESSIONAL_SANITARI;
grant select on HORAVISITA to PROFESSIONAL_SANITARI;
grant select on FITXA to PROFESSIONAL_SANITARI;
grant select on MEDICAMENT to PROFESSIONAL_SANITARI;
grant select on RECEPTA to PROFESSIONAL_SANITARI;
grant execute on INSERIR_FITXA to PROFESSIONAL_SANITARI;
grant execute on INSERIR_RECEPTA to PROFESSIONAL_SANITARI;
grant execute on MODIFICAR_FITXA to PROFESSIONAL_SANITARI;

-- Grant/Revoke system privileges
grant create session to PROFESSIONAL_SANITARI;
```

4. Rol ADMINISTRADOR_CONSULTORI

```
-- ROL AMINISTRADOR_CONSULTORI

-- Create the role
create role ADMINISTRADOR_CONSULTORI;

-- Grant/Revoke object privileges
grant select on PACIENT to ADMINISTRADOR_CONSULTORI;
grant select on HORAVISITA to ADMINISTRADOR_CONSULTORI;
grant select on PROFESSIONAL to ADMINISTRADOR_CONSULTORI;
grant select on MEDICAMENT to ADMINISTRADOR_CONSULTORI;
grant execute on ESBORRAR_HORAVISITA to ADMINISTRADOR_CONSULTORI;
grant execute on ESBORRAR_MEDICAMENT to ADMINISTRADOR_CONSULTORI;
grant execute on ESBORRAR_PACIENT to ADMINISTRADOR_CONSULTORI;
grant execute on ESBORRAR_PROFESSIONAL to ADMINISTRADOR_CONSULTORI;
grant execute on INSERIR_MEDICAMENT to ADMINISTRADOR_CONSULTORI;
grant execute on INSERIR_PROFESSIONAL to ADMINISTRADOR_CONSULTORI;
grant execute on MODIF_ESPECIALITATPROFESSIONAL to ADMINISTRADOR_CONSULTORI;
grant execute on CREAR_HORES_VISITA to ADMINISTRADOR_CONSULTORI;

-- Grant/Revoke role privileges
grant administratiu to ADMINISTRADOR_CONSULTORI;

-- Grant/Revoke system privileges
grant create session to ADMINISTRADOR_CONSULTORI;
grant create user to ADMINISTRADOR_CONSULTORI;
grant grant any role to ADMINISTRADOR_CONSULTORI;
grant drop user to ADMINISTRADOR_CONSULTORI;
```

5. CREACIÓ D'USUARIS

```
-- Creació d'USUARI

-- Create the user
create user UsuariProva
  identified by prova
  default tablespace USERS
  temporary tablespace TEMP;

-- Grant/Revoke role privileges
grant administratiu to UsuariProva;
grant professional_sanitari to UsuariProva;
grant administrador_consultori to UsuariProva;
```

5.- IMPLEMENTACIÓ

En aquesta etapa s'implementa el disseny físic. Es creen els *scripts* (codi) de creació dels objectes de la base de dades, així com els diferents rols i usuaris amb els permisos corresponents.

El codi executable es troba als arxius:

- Creació de taules, seqüències, paquet (*package*) i disparadors:
mpradera_tables_seq_pack_disparadors.sql

- Creació de procediments:
mpradera_procediments.sql

- Inserció dels paràmetres del consultori a la taula PARAMETRES:
mpradera_insercioParametresConsultori.pdc

- Creació de rols i usuaris:
mpradera_script_rols_i_usuaris.pdc

6.- PROVES

El joc de proves es troba a l'arxiu:
mpradera_annex2_proves.doc

I el codi executable:
mpradera_script_proves_procediments.pdc

7.- CONCLUSIONS

S'han assolit els objectius marcats. Desenvolupant aquest projecte informàtic a més de posar en pràctica els coneixements adquirits durant els estudis, n'he adquirit de nous mitjançant la utilització del sistema de gestió de bases de dades Oracle i el llenguatge de programació PL/SQL. Tots dos han representat un esforç important d'estudi.

La part més complicada ha estat el disseny i la implementació dels procediments, ja que a més de l'acció sobre les dades havien de complir uns requisits determinats i també s'havia de conèixer bé el llenguatge de programació.

Finalment, s'ha comprovat el bon funcionament del sistema gràcies al joc de proves.

8.- BIBLIOGRAFIA

William, G.(2001). *Oracle 8/8i*. Madrid: Prentice Hall.

Urman, S.(2002). *Oracle 9i Programación PL/SQL*. Madrid: McGraw-Hill.

Gabillaud, G.(2003). *Oracle 9i SQL, PL/SQL, SQL *Plus*. Barcelona: Ed. Software SL

Houlette, F. (2003). *Fundamentos de SQL*. Mèxic: McGraw-Hill.

Cuenca, M.J.; Marco, M.J.; Nicolau, F. (2005). *Competència comunicativa per a professionals de la informàtica*. Barcelona: Eureka Media SL.

Sistac, J. (2002). *Bases de dades I*. Barcelona: Eureka Media SL.

Sistac, J. (2004). *Bases de dades II*. Barcelona: Eureka Media SL.