

Treball Final de Carrera

*Gestió de la informació d'interès d'un
treballador autònom*

Berta Sañé Balasch

Enginyeria Tècnica d'Informàtica de Sistemes

Directora: M^a Dolors Anton Solà

Vic, setembre de 2007

ÍNDIX:

1.- Introducció _____	3
2.- Especificació de requeriments _____	4
3.- Anàlisi del sistema _____	5
3.1.- Model de dades _____	5
3.1.1.- Diagrama de classes _____	5
3.1.2.- Descripció del diagrama de classes _____	7
3.2.- Model funcional _____	10
3.2.1.- Descripció d'usuari _____	10
3.2.2.- Llista d'esdeveniments _____	12
4.- Disseny del sistema _____	18
4.1.- Disseny d'interfícies en l'entorn personal _____	19
4.2.- Disseny d'interfícies en l'entorn web _____	39
4.3.- Disseny de la base de dades _____	52
4.3.1.- Descripció de les taules _____	54
4.4.- Disseny de programes _____	57
4.4.1.- Disseny de l'entorn personal _____	57
4.4.2.- Disseny de l'entorn web _____	63
5.- Disseny arquitectònic i manual d'instal·lació _____	68
6.- Millores i conclusions _____	72
7.- Bibliografia _____	73

1.- INTRODUCCIÓ:

L'objectiu del projecte que tenim a continuació, és de realitzar una aplicació per a poder gestionar la informació d'un treballador autònom. Per a tal de portar a terme això, hem creat una aplicació per portar el control de totes les dades (altes, consultes i modificacions), i una pàgina web, a la qual només es pot accedir amb un nom d'usuari i una contrasenya, on es podran fer tota mena de consultes, a més de portar l'agenda del centre i la del treballador. A més de tot el que hem esmentat anteriorment, s'ha de poder fer un control dels ingressos de les diferents empreses, i les factures necessàries.

2.- ESPECIFICACIÓ DE REQUERIMENTS:

L'objectiu que volem assolir amb la realització d'aquest projecte és la gestió de la informació d'interès d'un treballador autònom. Bàsicament aquesta empresa es dedica al món de l'estètica i l'interessa poder realitzar un bon seguiment dels clients i portar la facturació. L'empresa consta de dos centres actualment, els quals han de poder accedir a la base de dades sense problemes. El treballador autònom li interessa poder tenir la informació dels dos centres i poder-ne fer un seguiment.

S'hauria de realitzar una base de dades on i figurin tots els clients, cal destacar que en podem tenir dos tipus de clients: personals i centres (altres empreses). En quan a la fitxa de cada client, haurà de comptar amb les dades bàsiques de la persona, com és el nom, cognoms, adreça, telèfon de contacte, etc. A més, tindrem un historial del client, on hi figurarà el tractament o tractaments que s'estigui realitzant, juntament amb les potències tractades i el dia de cada sessió, i el preu que li correspon. Ens interessaria saber si es queda algun producte dels que ofereix l'empresa. Per últim els clients s'han de poder ordenar (segons conveniència), per nom, zona tractada, data d'inici del tractament...

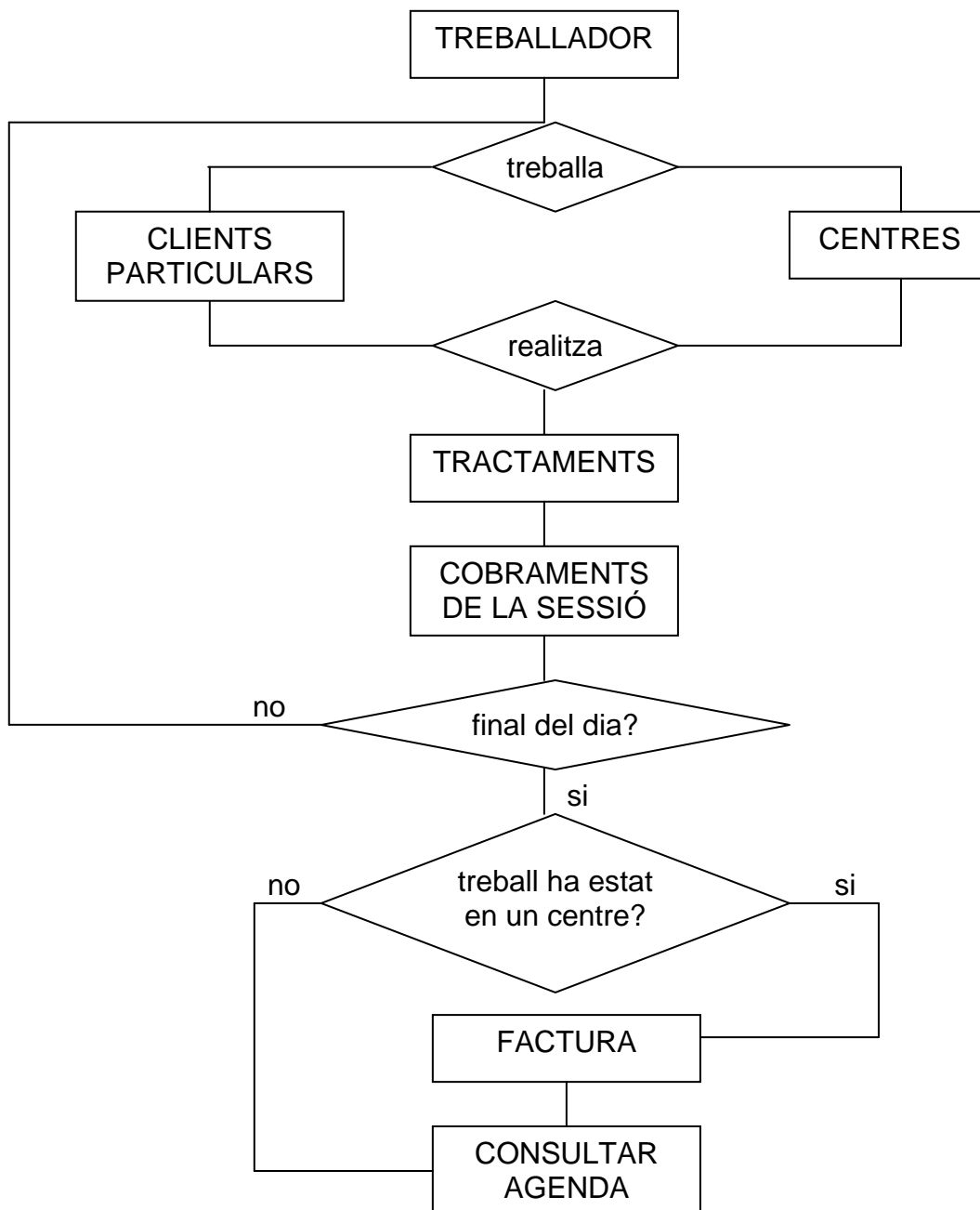
També tindrem informació dels productes que l'empresa ofereix, i aquests estaran relacionats amb el seu proveïdor, dels quals ens interessa guardar-ne les dades.

Fent referència a la facturació, interessa que els ingressos es vagin comptabilitzant automàticament una vegada cobrada la sessió. També s'ha de poder veure la facturació que hi ha hagut en un període de temps especificat per l'usuari. Per últim ens ha de permetre poder fer les factures de cada mes per a les diferents empreses que tenim, s'ha de tenir en compte el percentatge que guanya l'empresa del total de la facturació. (Cal dir que tot el tema de facturació el vol per ús personal).

Finalment a l'empresa li és interessant tenir una agenda detallada per saber els clients que li venen durant el dia, o per saber en el centre on treballarà.

Tot seguit s'explicarà de forma detallada el que efectua el treballador autònom en un dia de treball. Cal diferenciar si el treball és particular o en una altra empresa. En primer lloc explicarem les tasques a realitzar en cas de que el treball sigui particular, en començar el dia, obre l'agenda per saber la feina que es durà a terme durant el dia, on veu les persones que van a fer-se tractaments, i el tractament que es realitza cada una d'elles. En el moment que té un client, obre la seva fitxa personal amb tot l'historial, on mira quan es va fer la última sessió i les potències que va utilitzar, una vegada fet això es fa el tractament, i quan l'ha finalitzat es cobra i es queda per el dia de la pròxima sessió (on cal consultar l'agenda). Això es va repetint amb tots els

clients fins al final del dia, finalment cal consultar novament l'agenda per poder veure els clients que han d'assistir-hi l'endemà per tal de poder-los avisar via mòbil. En el cas de que el treball es porti a terme en una altre empresa, el que fa és el següent: es desplaça a l'empresa on ha de treballar, una vegada allà seguirà el procediment explicat anteriorment fins al cobrament del tractament, en finalitzar el dia es farà la factura corresponent al dia de feina amb el tant per cent que es queda l'empresa. Finalment cal consultar novament l'agenda per informar els clients de dia següent. Recordem que existeixen varies seus, des d'on s'ha de poder treballar, ja que l'empresa consta de dos establiments, i a més, treballa a diferents centres.



3.- ANÀLISI DEL SISTEMA:

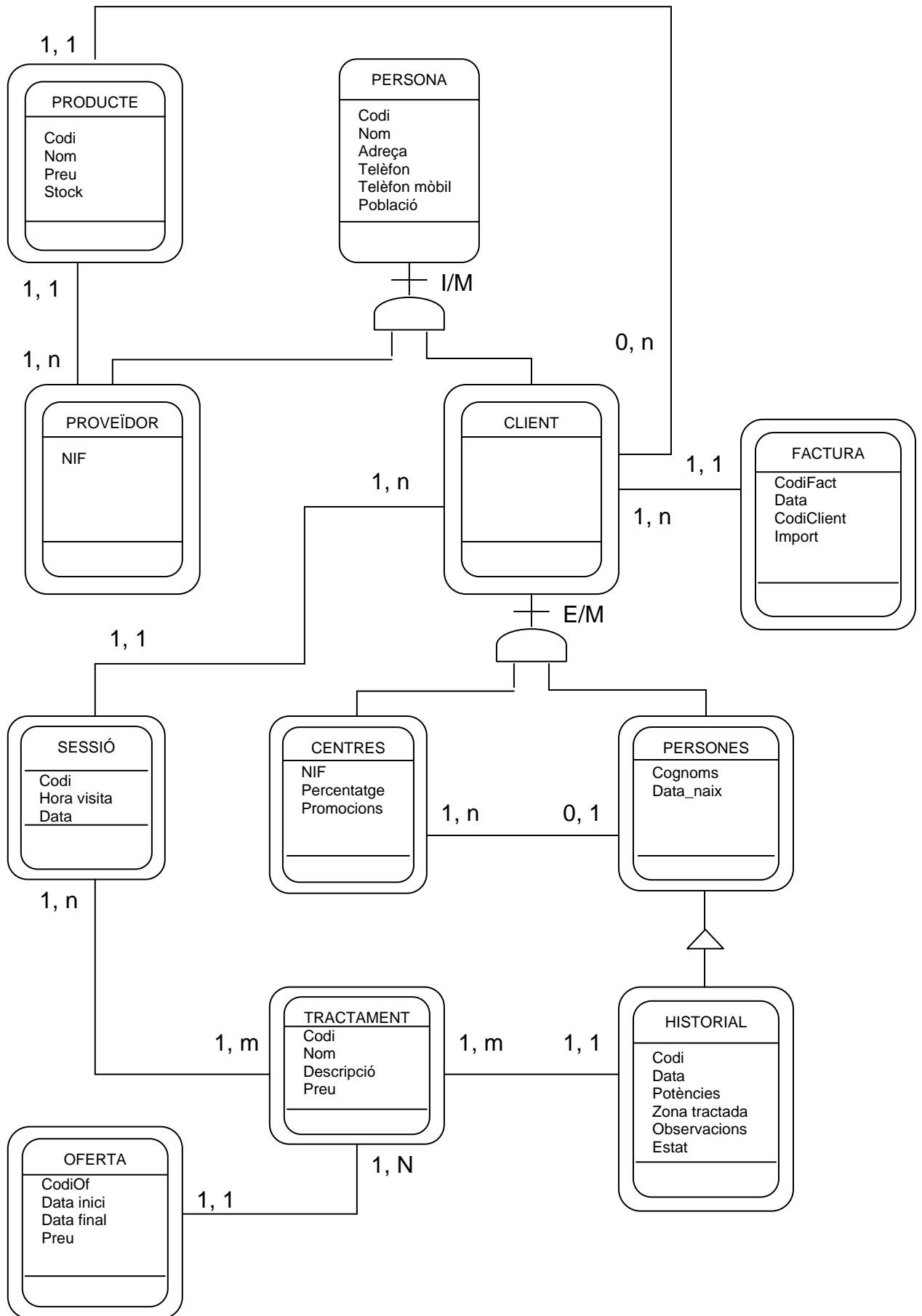
L'anàlisi del sistema és una fase a seguir per tal de desenvolupar un projecte informàtic. Una vegada obtinguda la solució conceptual del problema, especificarem què és el que ha de fer el sistema sense lligar-ho a cap tecnologia concreta, com a resultat hem d'obtenir les funcions que ha de fer el sistema, definir el flux d'informació que circula al llarg del sistema i la informació que manipula el sistema. Fent l'anàlisi es generen models: el model de dades i el model funcional. El model de dades és el punt de vista estàtic del sistema que permet veure els elements dels quals, al sistema l'interessa guardar la informació i la relació entre ells. D'altra banda, el model funcional és un punt de vista dinàmic del sistema, que permet veure quines són les funcions que ha de fer, quins esdeveniments ha de detectar i com ha de respondre davant d'ells. La metodologia que utilitzarem per a fer-ho és la metodologia de Yourdon orientada a objectes.

3.1.- Model de dades:

Per tal de desenvolupar el model de dades, crearem un diagrama de classes i seguidament el descriurem.

3.1.1.- Diagrama de classes:

Un diagrama de classes és una representació gràfica d'un conjunt d'objectes. El diagrama de classes és el diagrama principal per a fer l'anàlisi i el disseny. Aquest presenta les diferents classes del sistema, que representen objectes d'interès del domini i les seves propietats, amb les seves relacions estructurals.



3.1.2.- Descripció del diagrama de classes:

Una vegada fet el diagrama de classes, per completar-ho cal documentar-lo. Això ens permet saber més informació sobre els atributs de les classes i dels seus serveis. Seguidament es farà una petita descripció de les diferents classes que hi ha en el diagrama de classes.

PERSONA: classe necessària per tal de fer la generalització entre els tipus de persones que podem trobar, clients i proveïdors. Els atributs que tindrà la classe persona són:

- Codi: Numèric
- Nom: Alfabètic
- Adreça: Alfabètic
- Telèfon: Alfabètic
- Mòbil: Alfabètic
- Població: Alfabètic

El moment de fer l'especificació d'aquesta classe surten els símbols I/M. El significat d'aquestes lletres signifiquen inclusiva/exhaustiva, inclusiva perquè una persona pot ser alhora proveïdor i client, i exhaustiva perquè és una classificació completa, és a dir només hi poden haver dos tipus de persones, clients i proveïdors.

PROVEÏDOR: necessària per guardar la informació dels proveïdors, la informació que ens interessa és:

- *Hereta de PERSONA
- NIF: Numèric

PRODUCTE: aquesta classe s'utilitzarà per definir els productes que ofereix l'empresa, els seus atributs són els següents:

- CodiProd: Numèric
- Nom: Alfabètic
- Preu: Numèric
- Stock: Numèric

CLIENT: és una classe que utilitzem per a fer la generalització dels tipus de clients que podem tenir. Aquesta classe no tindrà cap atribut afegit, només els que hereta de PERSONA.

En el moment de fer l'especificació de la classe CLIENT, apareix el símbol E/M. E, perquè es una classificació exclusiva, és a dir un client no pot ser a la vegada client de centre i client personal i viceversa, i M perquè és una classificació completa, és a dir només hi poden haver dos tipus de clients.

CENTRES: és la classe que farem servir per a guardar la informació dels centres amb els quals hi tenim clients.

*Hereta de CLIENT

- NIF: Numèric
- Percentatge: Numèric
- Promocions: Alfabètic

PERSONES: en aquesta classe guardarem la informació dels clients, persones físiques, les quals han accedit a nosaltres per realitzar-se algun dels nostres tractaments.

*Hereta de CLIENT

- Cognoms: Alfabètic
- Data naix: Date

FACTURA:

- CodiFac: Numèric
- CodiCentre: Numèric
- Data: Date
- Import total: Numèric
- D_ini: Date
- D_fi: Date

SESSIÓ: classe on guardarem la informació més rellevant, com és la hora de visita d'una sessió i el tractament o tractaments que s'han de fer els clients en aquella sessió.

- Codi: Numèric
- Hora: Time
- Data: Date

HISTORIAL: aquí guardarem tot l'historial del client, on ens ensenyarà els tractament que es fa un client en concret i un seguiment de cada tractament on posarem la data i les potències utilitzades en cada sessió.

- Codi: Numèric
- Data: Data
- Potències: Numèric
- Zona tractada: Alfabètic
- Observacions: Alfabètic
- Estat: Alfabètic

TRACTAMENT: en aquesta classe definirem els diferents tipus de tractament que podem realitzar, els seus atributs seran els següents:

- CodiTrac: Numèric
- Nom: Alfabètic
- Descripció: Alfabètic
- Preu: Numèric

OFERTA: en aquesta classe es defineixen les diferents ofertes que es poden donar pels tractament que podem realitzar, els seus atributs seran els següents:

- CodiOf: Numèric
- NomTrac: Alfabètic
- Data inici: Date
- Data final: Date
- Preu: Numèric

Sabem que els objectes tenen dos tipus de propietats: els atributs i els serveis. En l'especificació feta fins el moment podem observar que només hi ha definits els atributs, això és perquè encara no s'han detectat els serveis, ja que es detecten en el moment de descriure l'aspecte dinàmic del sistema, punt que es farà tot seguit en el model funcional.

3.2.- Model funcional:

En el model funcional es descriuran les funcions que ha de dur a terme el sistema des del punt de vista d'usuari, a més dels esdeveniments que se'n deriven.

Per tal de generar aquest model es seguiran els següents passos:

- Descripció d'usuaris.
- Llista d'esdeveniments.
- Ampliació del diagrama de classes.

3.2.1.- Descripció d'usuari:

Cal dir que la descripció dels usuaris és important per poder decidir més endavant, quines seran les interfícies que formaran part del sistema i deixaran comunicar el sistema amb l'entorn. És important també, per descobrir quines són les tasques que porta a terme i que suposaran esdeveniments que el sistema haurà de detectar.

Nom usuari: Autònom.

Tipus usuari: Regular ja que utilitzarà habitualment l'aplicació i de manera estructurada.

Relació feina / sistema: El sistema s'utilitzarà per guardar totes les dades i historials dels clients. La feina de l'autònom consisteix en donar detall de tots els tractaments que es fan a un pacient, i ha de poder entrar i/o anul·lar les visites. A més ha de poder fer factures per al centre, i veure els ingressos que hagi experimentat amb un període determinat. Finalment ha de poder portar el dia les agendes (la del centre, i la seva).

Necessitats i requeriments: Per un costat, ha de poder fer altes de clients, a més de poder consultar les seves dades i el seu historial, la forma de buscar la informació a de ser el més senzilla possible. L'entrada de dades ha d'estar simplificada el màxim. Una altra necessitat, és poder entrar visites, s'ha d'especificar el nom del client i el tractament que es fa aquell dia. Finalment ha de poder fer factures per els centres.

Forma de treballar: L'usuari entrarà a l'aplicació i podrà escollir quina opció necessita fer, si altes, consultes... Quan tingui una visita, entrarà a la fitxa del client, on mirerà l'historial, una vegada finalitzada la sessió, actualitzarà la informació. Per tal d'assignar visites es tindrà en compte la disponibilitat del client i la durada del tractament. Si un pacient demana un canvi el procediment que es segueix és esborrar la visita antiga i afegir la nova visita. Quan entri a centres, hi haurà l'apartat de factures, on demanarà l'interval de temps del qual s'ha de fer la factura i a continuació es farà.

Nom usuari: Treballador del centre associat.

Tipus usuari: Regular ja que utilitzarà habitualment l'aplicació i de manera estructurada.

Relació feina / sistema: El sistema s'utilitzarà per guardar totes les dades i historials dels clients. La feina consisteix en donar detall de tots els tractaments que es fan a un pacient. Finalment ha de poder consultar les agendes en tot moment, i fer-ne els canvis que convinguin.

Necessitats i requeriments: Ha de poder fer altes de clients, a més de poder consultar les seves dades i el seu historial, la forma de buscar la informació a de ser el més senzilla possible. L'entrada de dades ha d'estar simplificada el màxim.

Forma de treballar: L'usuari entrarà a l'aplicació i podrà escollir entre clients o agenda. Quan tingui una visita, entrarà a la fitxa del client, on mirerà l'historial, una vegada finalitzada la sessió, actualitzarà la informació.

3.2.2.- Llista d'esdeveniments:

En aquest punt descriurem tots els esdeveniments tenint en compte l'especificació i les tasques a desenvolupar per cada usuari. Recordem que un esdeveniment és quelcom pot passar a l'entorn del sistema, que aquest ho ha de detectar i hi ha de donar resposta. Per tal d'especificar els esdeveniments es seguirà la següent estructura:

- **Esdeveniment:** nom de l'esdeveniment
- **Tipus:** de quin tipus d'esdeveniment es tracta
- **Resposta:** la resposta que ha de donar el sistema
- **Descripció:** descripció de l'esdeveniment
- **Serveis:** serveis que s'hauran d'executar per tal de donar resposta

Esdeveniment: Alta de centre

Tipus: Informació

Resposta: Es dona d'alta un nou centre

Descripció: Rebre les dades d'un nou centre, comprovar el NIF i gravar-les a la base de dades.

Serveis: Centre.Consultar, Centre.Crear

Esdeveniment: Buscar centre

Tipus: Informació

Resposta: Es consulten les dades d'un centre

Descripció: Consultar i mostrar les dades del centre seleccionat.

Serveis: Centre.Consultar

Esdeveniment: Modificar centre

Tipus: Informació

Resposta: Es modifiquen les dades d'un centre

Descripció: Consultar i mostrar les dades del centre seleccionat, es modifiquen i es guarden.

Serveis: Centre.Consultar, Centre.Modificar

Esdeveniment: Alta de Persones

Tipus: Informació

Resposta: Es dona d'alta un nou client

Descripció: Rebre les dades d'un nou client, comprovar el DNI i gravar-les a la base de dades.

Serveis: Persones.Consultar, Centre.Consultar, Persones.Crear

Esdeveniment: Buscar Persones

Tipus: Informació

Resposta: Es busquen les dades d'un client

Descripció: Consultar i mostrar les dades del client seleccionat.

Serveis: Persones.Consultar

Esdeveniment: Modificar Persones

Tipus: Informació

Resposta: Es modifiquen les dades d'un client

Descripció: Consultar i mostrar les dades del client seleccionat, es modifiquen i es guarden.

Serveis: Persones.Consultar, Centre.Consultar, Persones.Modificar

Esdeveniment: Alta de proveïdor

Tipus: Informació

Resposta: Es dona d'alta un nou proveïdor

Descripció: Rebre les dades d'un nou proveïdor, comprovar el NIF i gravar-les a la base de dades.

Serveis: Proveidor.Consultar, Proveidor.Crear

Esdeveniment: Modificar proveïdor

Tipus: Informació

Resposta: Es modifiquen les dades d'un proveïdor

Descripció: Consultar i mostrar les dades del proveïdor seleccionat, es modifiquen i es guarden.

Serveis: Proveidor.Consultar, Proveidor.Modificar

Esdeveniment: Buscar proveïdor

Tipus: Informació

Resposta: Es mostren les dades del proveïdor

Descripció: S'entra el nom del proveïdor i surten les seves dades.

Serveis: Proveidor.Consultar

Esdeveniment: Alta de Producte

Tipus: Informació

Resposta: Es dona d'alta un nou producte

Descripció: Rebre les dades del producte, i les grava.

Serveis: Producte.Consultar, Proveïdor.Consultar, Producte.Crear

Esdeveniment: Buscar Producte

Tipus: Informació

Resposta: Es busca la informació d'un producte

Descripció: Mostrar les dades del producte.

Serveis: Producte.Consultar

Esdeveniment: Modificar Producte

Tipus: Informació

Resposta: Es modifica la informació d'un producte

Descripció: Modificar i guardar les dades d'un producte.

Serveis: Producte.Consultar, Proveïdor.Consultar, Producte.Modificar

Esdeveniment: Alta de Tractament

Tipus: Informació

Resposta: Es dona d'alta un tractament

Descripció: Guardar les dades d'un tractament.

Serveis: Tractament.Consultar, Tractament.Crear

Esdeveniment: Buscar Tractament

Tipus: Informació

Resposta: Es busca la informació d'un tractament

Descripció: Mostrar les dades del tractament.

Serveis: Tractament.Consultar

Esdeveniment: Modificar Tractament

Tipus: Informació

Resposta: Es modifica la informació d'un tractament

Descripció: Modificar i guardar les dades d'un tractament.

Serveis: Tractament.Consultar, Tractament.Modificar

Esdeveniment: Consultar Sessió

Tipus: Control

Resposta: Es mostraran els clients, la hora de visita i el tractament a realitzar del dia seleccionat.

Descripció: Es busquen les visites del dia i es mostren per pantalla.

Serveis: Sessió.Consultar

Esdeveniment: Afegir Sessió

Tipus: Informació

Resposta: Afegir una nova visita a l'agenda.

Descripció: Es busquen les visites ja concretades, i s'incorpora la nova visita en una hora que no hi hagi ningú.

Serveis: Sessió.Consultar, Persones.Consultar, Tractament.Consultar, Sessió.Afegir

Esdeveniment: Esborrar Sessió

Tipus: Informació

Resposta: S'esborra el dia i la hora de la visita a l'agenda.

Descripció: Es busca la visita concretada i s'esborra.

Serveis: Sessió.Consultar, Sessió.Borrar

Esdeveniment: Afegir historial

Tipus: Informació

Resposta: Es guarden les noves dades a l'historial

Descripció: S'escriu el tractament i les observacions de la visita i es guarda en el seu historial.

Serveis: Tractament.Consultar, Historial.Afegir

Esdeveniment: Consultar Ingressos

Tipus: Control

Resposta: Es mostren els ingressos que hi han hagut en un període de temps determinat.

Descripció: Es selecciona la data inici i la data final per definir l'interval de temps, i es mostra els ingressos en aquell període.

Serveis: Factura.Consultar, Client.Consultar, Sessió. Consultar_per_dates, Tractament.consultar

Esdeveniment: Fer Factura

Tipus: Informació

Resposta: Es mostren la factura del centre escollit, hi es pot imprimir.

Descripció: S'escull el centre del qual s'ha de fer la factura i s'imprimeix.

Serveis: Client.Consultar, Sessio.Consultar_client, Tractament.Consultar, Factura.Crear, Factura.Imprimir

Esdeveniment: Alta d'usuari

Tipus: Informació

Resposta: Es dona d'alta un nou usuari.

Descripció: Guarda les dades (nom i contrasenya) d'un nou usuari.

Serveis: Usuari.Consultar, Usuari.Crear

Esdeveniment: Baixa usuari

Tipus: Informació

Resposta: Es busca la informació d'un usuari i s'esborra.

Descripció: Mostrar les dades de l'usuari i seguidament les esborra..

Serveis: Usuari.Consultar, Usuari.Esborrar

Esdeveniment: Alta oferta

Tipus: Informació

Resposta: Es dona d'alta una oferta d'un tractament

Descripció: Guarda a la base de dades les dades de la oferta.

Serveis: Oferta.Consultar, Tractament.Consultar, Oferta.Crear

Esdeveniment: Baixa oferta

Tipus: Informació

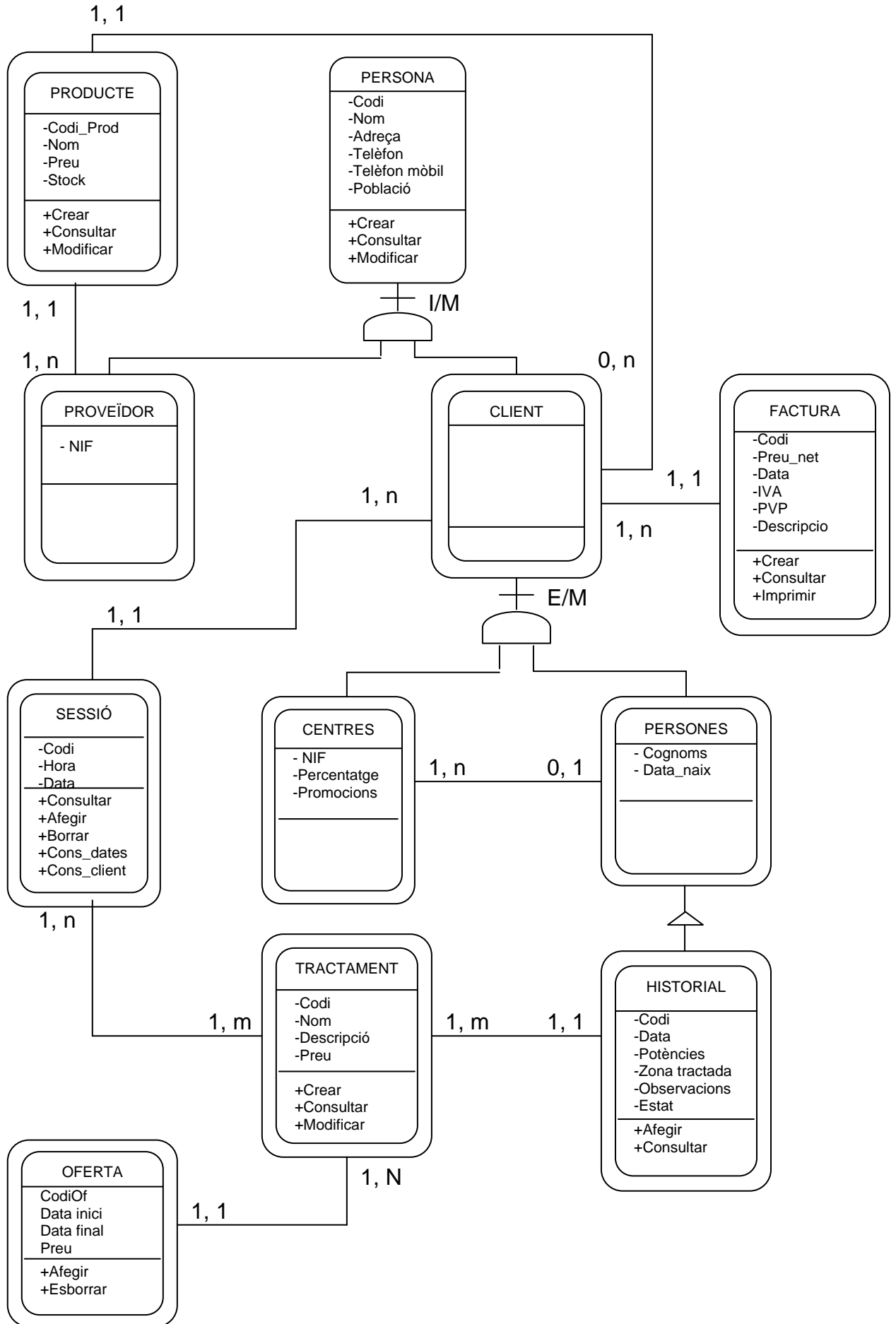
Resposta: Es dona de baixa de la base de dades una oferta.

Descripció: Mostrar les dades d'una oferta i seguidament les esborra.

Serveis: Oferta.Consultar, Oferta.Esborrar

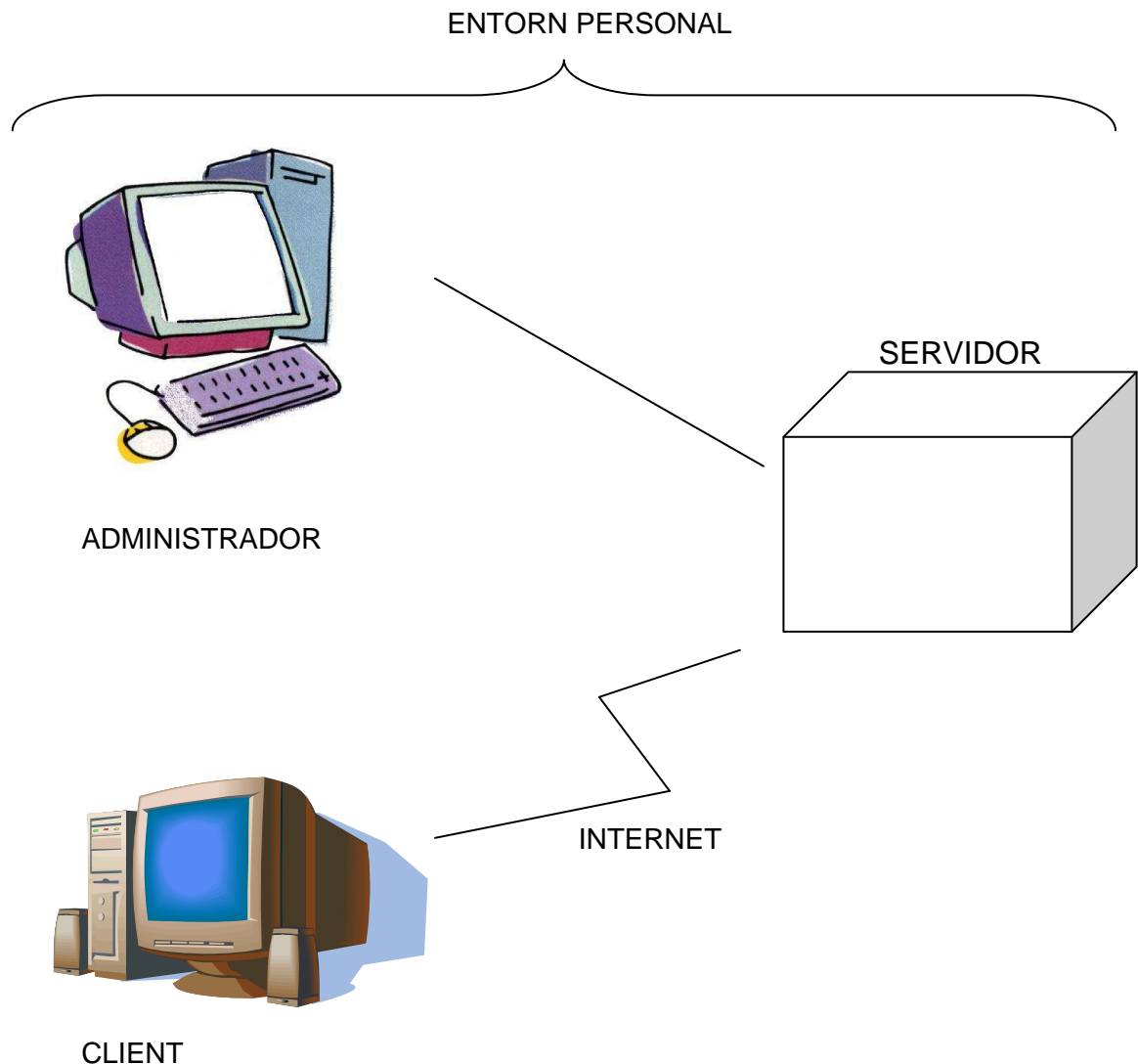
Es pot observar una vegada anomenats tots els esdeveniments, que no hi consta cap baixa (centres, persones, productes i tractaments). La raó per a la qual no hi són, és que la persona interessada amb el programa no li eren útils, ja que li interessa guardar tota la informació encara que no ho hagi d'utilitzar.

Una vegada vistos els esdeveniments que es poden produir en el sistema, ens adonem que cal realitzar una ampliació del diagrama de classes, ja que hem obtingut noves propietats de les classes, els serveis.



4.- DISSENY DEL SISTEMA:

El disseny del sistema consisteix en orientar l'anàlisi cap a una implementació, és a dir, lligar el resultat de l'anàlisi a una tecnologia concreta. En el nostre cas, l'aplicació es divideix amb dues parts: una és un entorn personal, i l'altra part un entorn web. L'entorn personal l'utilitzarem per tal de que el treballador autònom, que és l'encarregat de gestionar totes les dades, pugui fer altes, modificacions i consultes de tot tipus de dades. Cal que recordem que és l'únic que pot accedir a modificar i fer altres de noves dades. I finalment com que hi ha d'haver varis punts des don t'has de poder connectar al servidor per tal de poder fer consultes, ja que comptem amb més d'una seu, per això és necessari l'entorn web, que ens permetrà connectar-nos en tot moment a la base de dades i consultar qualsevol dada sempre en funció dels privilegis de l'usuari.



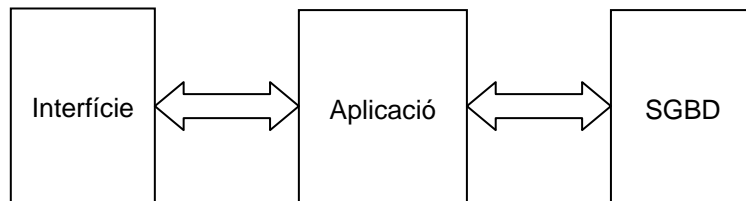
Independentment de l'entorn d'explotació per fer el disseny cal seguir els següents punts:

- Disseny de les interfícies de comunicació entre l'usuari i la màquina.
- Disseny de la base de dades.
- Disseny del programa.

Al ser entorns diferents es descriuen a continuació els punts per cada un d'ells.

4.1.- Disseny d'interfícies en l'entorn personal:

L'estructura de l'entorn personal que nosaltres disposem, està compost per: interfícies, l'aplicació, i finalment la base de dades, totes elles estan connectades tal i com es mostra en el següent esquema:

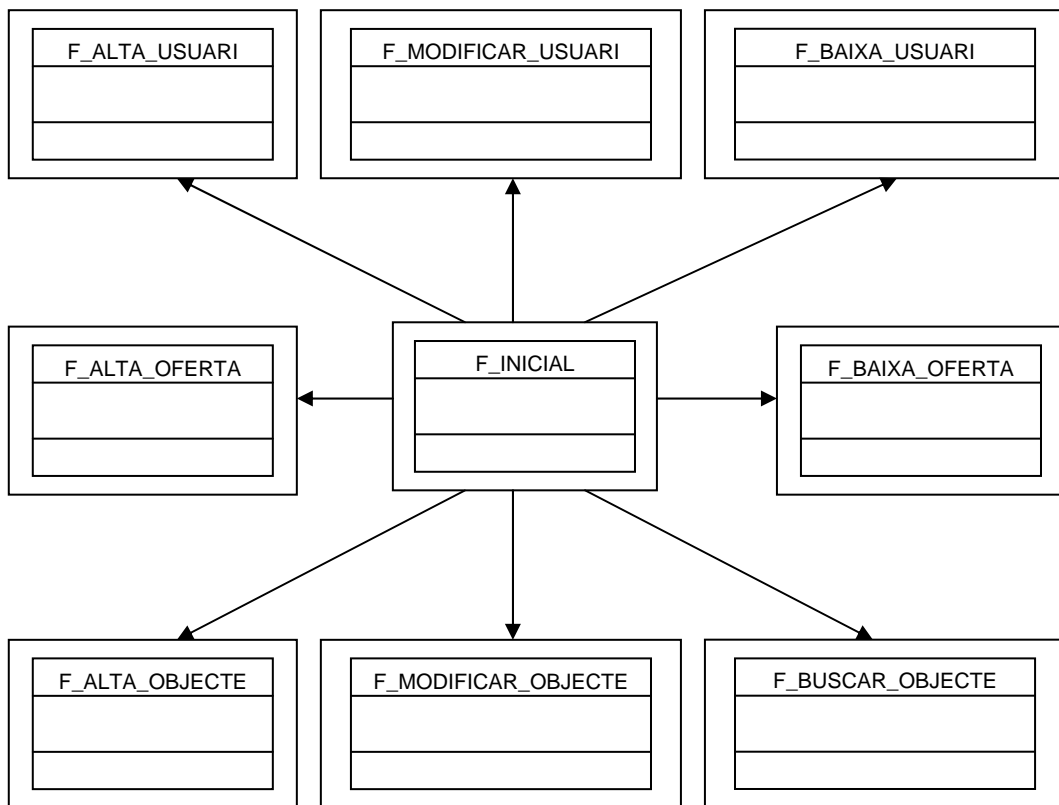
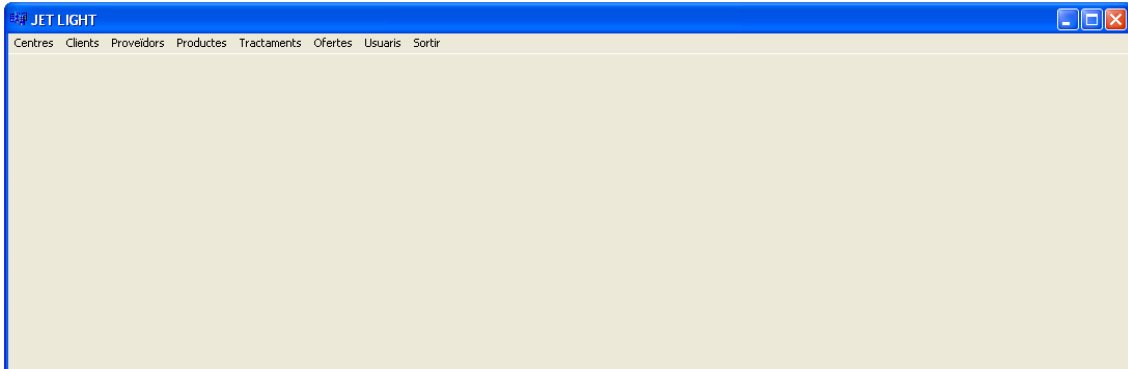


En l'apartat anterior hem descrit tots els possibles esdeveniments que hi poden haver en el sistema. Resulta, que cada esdeveniment necessita una interfície per tal de poder comunicar l'usuari amb la màquina. Aquesta interfície és un nou objecte d'interès, que serà representat de manera abstracte com una classe. Aquesta nova classe ha d'actuar com a classe controladora, amb això vull dir que s'ha de cuidar de fer les crides necessàries als serveis dels objectes per tal de dur a terme la resposta a l'esdeveniment produït en ella. S'ha de tenir en compte que els objectes, llevat que siguin dependents no es passen missatges entre ells, cosa que implica una bona encapsulació.

Ara veurem les diferents interfícies de comunicació amb l'usuari dissenyades i les seves relacions amb les diferents classes definides en el diagrama de classes.

F_INICIAL:

Finestra inicial per la gestió de l'usuari. Des d'aquí podrà escollir la part de l'aplicació on vol accedir. La paraula OBJECTE que es veu a continuació es pot intercanviar per a qualsevol d'aquestes PERSONES, CENTRES, PROVEÏDOR, PRODUCTE i TRACTAMENT, ja que són classes que fan els mateixos esdeveniments.



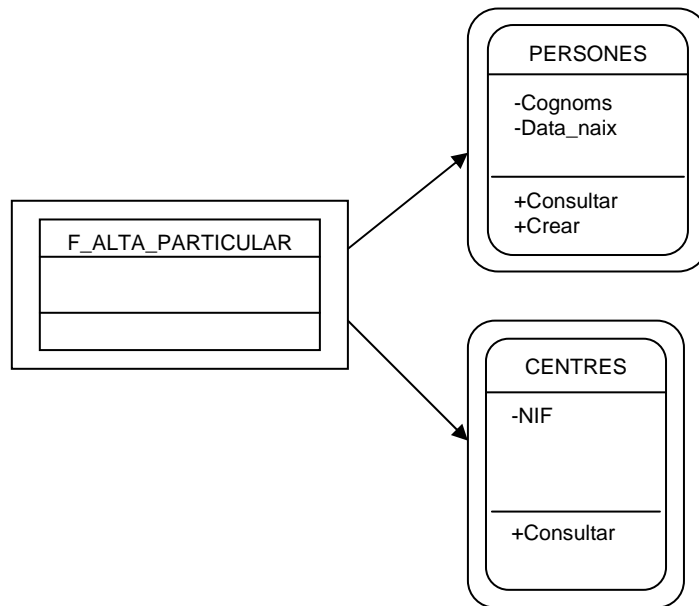
F_ALTA_PERS:

Finestra on l'usuari podrà donar d'alta els clients personals, posant totes les seves dades.

The screenshot shows a window titled 'ALTA CLIENT' with a light beige background. The form contains the following fields and values:

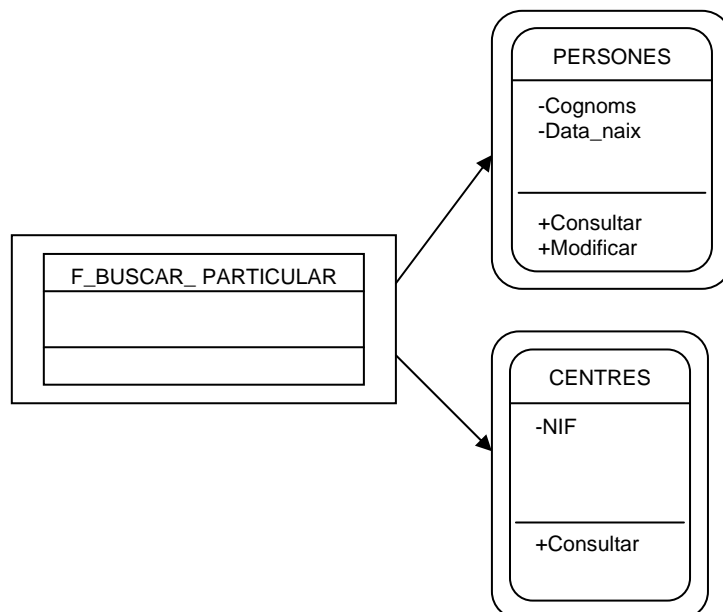
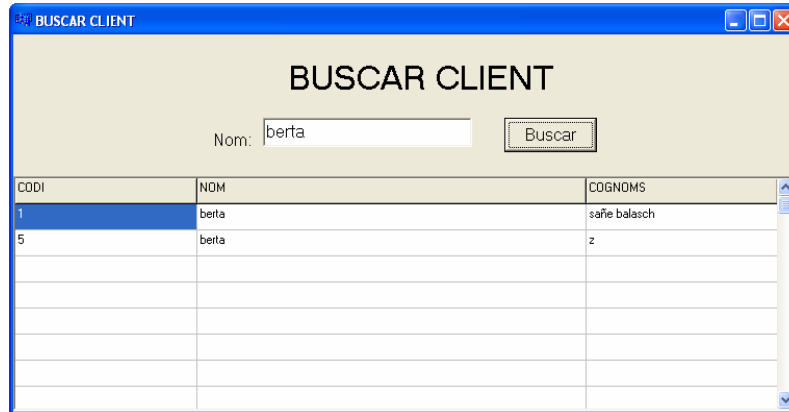
- Codi: 2
- Nom: Berta
- Cognoms: Safé Balasch
- Data Naixement: 16/03/1985 (dropdown menu)
- Adreça: Escorial 6
- Població: Vic
- Telèfon: 93 889 05 19
- Telf. Mobil: 699 36 79 89
- Centre associat: Jet Light

Buttons: 'Donar alta' and 'Sortir'.



F_BUSCAR_PERS:

Finestra a la qual s'accedirà per tal de buscar la informació d'un client.



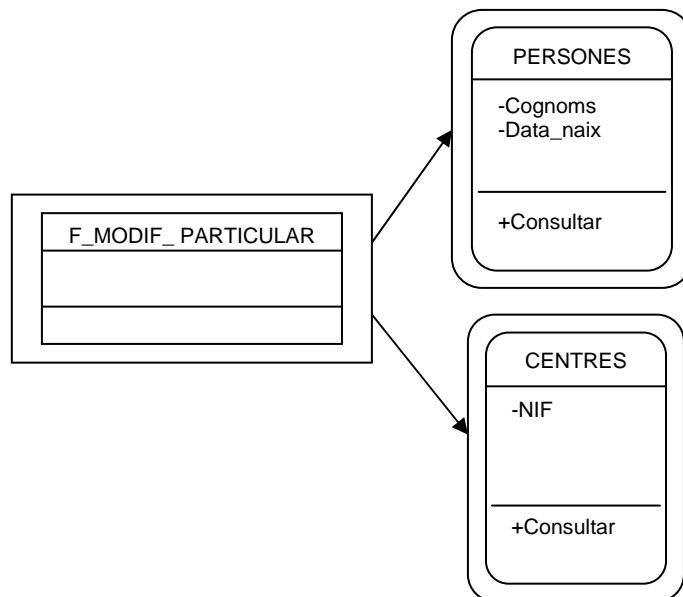
F_MODIFICAR_PERS:

Finestra des d'on buscarem i modificarem la informació d'un client.

CODI	NOM	COGNOMS
2	Berta	Sañé Balasch
3	Berta	Lopez Farres

Codi: 2
Nom: **Berta**
Cognoms: **Sañé Balasch**
Data Naixament: **16/03/1985**
Adreça: Escorial 6
Població: Vic
Telèfon: 93 889 05 19
Telf. Mobil: 699 36 79 89
Centre associat: **Jet Light**

Modificar
Sortir



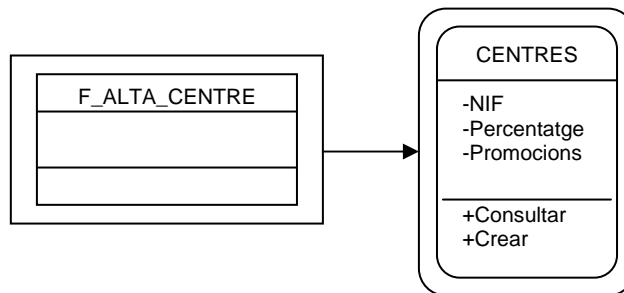
F_ALTA_CENTRE:

Finestra on l'usuari podrà donar d'alta els centres, posant totes les seves dades.

The screenshot shows a web browser window titled 'ALTA CENTRE'. The form contains the following fields and controls:

- Codi: 4
- Nom: Prova
- NIF: 12345678-
- Adreça: esfrfr
- Població: vic
- Telefon: 93 888 88 88
- Telf. Mobil: 666 66 66 66
- Percentage: 20 %
- Promocions: Si No
- Buttons: Donar alta, Sortir

An overlay message box titled 'Entrada correcta' is displayed, containing the text 'Centre introduït correctament a la base de dades' and an 'Acceptar' button.



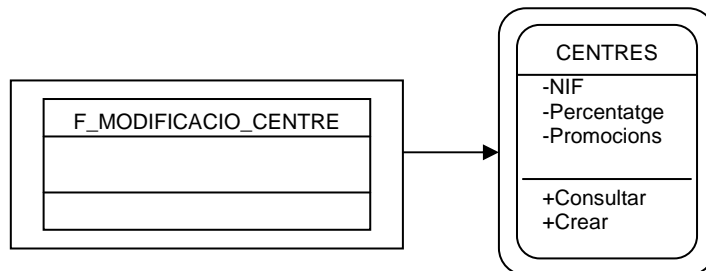
F_MODIFICAR_CENTRE:

Finestra a la qual s'accedirà per tal de modificar la informació d'un centre.

The screenshot shows a web application window titled "MODIFICAR CENTRE". The window contains a form with the following fields and values:

- Codi: 4
- Nom: Prova
- NIF: 12345678-L
- Adreça: prova adreça
- Població: vic
- Telefon: 93 888 88 88
- Telf. Mobil: 666 66 66 66
- Percentatge: 20 %
- Promoció: Si No

Buttons visible in the window include "Buscar", "Modificar", and "Sortir". A small dialog box titled "Modificació" is open, showing the message "Centre modificat correctament" and an "Acceptar" button.

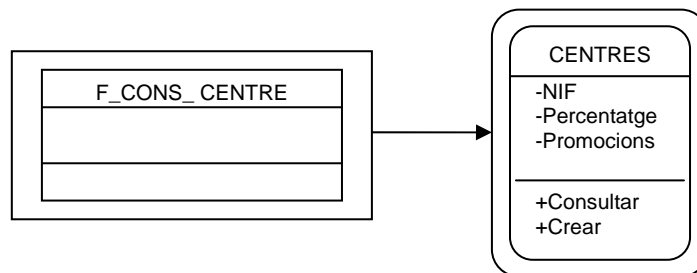


F_BUSCAR_CENTRE:

Finestra des d'on buscarem un centre per tal de veure les seves dades.

The screenshot shows a window titled "BUSCAR CENTRE" with a light beige background and a blue border. The window contains the following text and controls:

- Codi:** 1
- Nom:** Prova (with a text input field containing "Prova" and a "Buscar" button to its right)
- NIF:** 333
- Adreça:** prova adreça
- Població:** ee
- Telefon:** 93 888 88 88
- Telf. Mobil:** 666 66 66 66
- Percentage:** 20 %
- Promocions:** No
- A "Sortir" button is located at the bottom right of the window.



F_ALTA_PROV:

Finestra on l'usuari podrà donar d'alta els proveïdors, posant totes les seves dades.

ALTA PROVEIDOR

Codi: 2

Nom:

NIF:

Adreça:

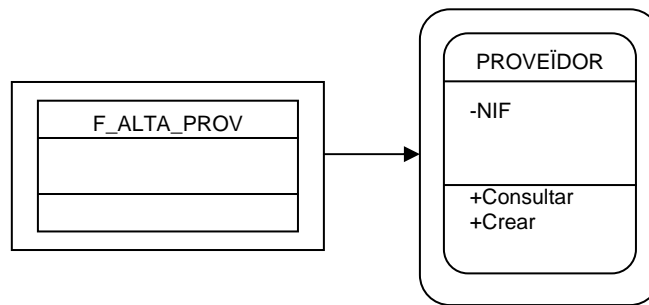
Població:

Telefon:

Telf. Mobil:

Donar alta

Sortir



F_MODIFICAR_PROV:

Finestra a la qual s'accedirà per tal de poder modificar la informació d'un proveïdor.

MODIFICAR PROVEIDOR

Codi:

Nom:

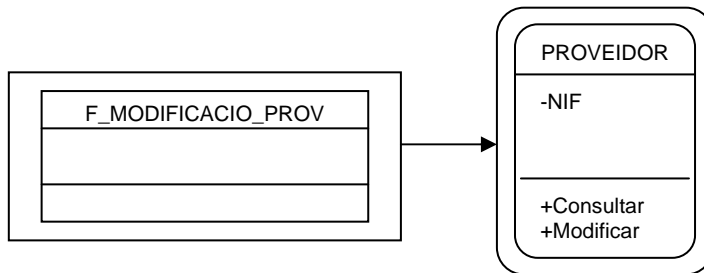
NIF:

Adreça:

Poblacio:

Telefon:

Telf. Mobil:



F_BUSCAR_PROV:

Finestra des d'on buscarem les dades d'un proveïdor.

BUSCAR PROVEIDOR

Codi:

Nom:

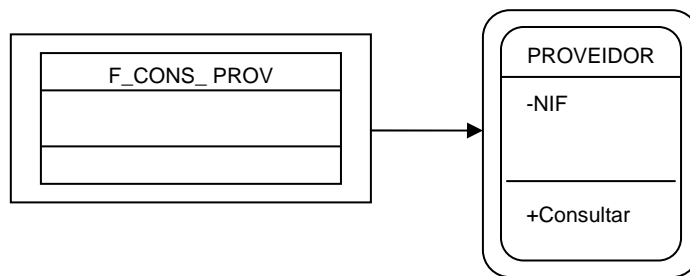
NIF:

Adreça:

Població:

Telefon:

Telf. Mobil:



F_ALTA_PROD:

Finestra on l'usuari podrà donar d'alta els productes.

ALTA PRODUCTE

Codi:

Nom:

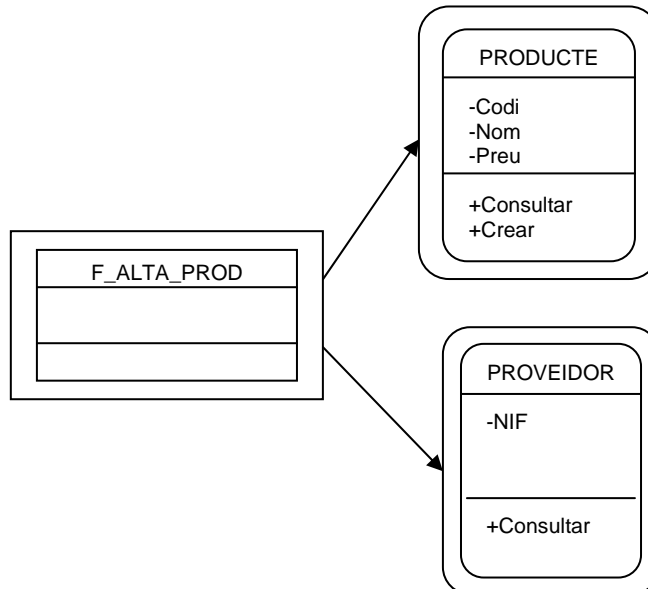
Preu: €

Stock:

Nom proveïdor:

Donar Alta

Sortir



F_MODIFICAR_PROD:

Finestra a la qual s'accedirà per tal de poder modificar la informació d'un producte.

MODIFICAR PRODUCTE

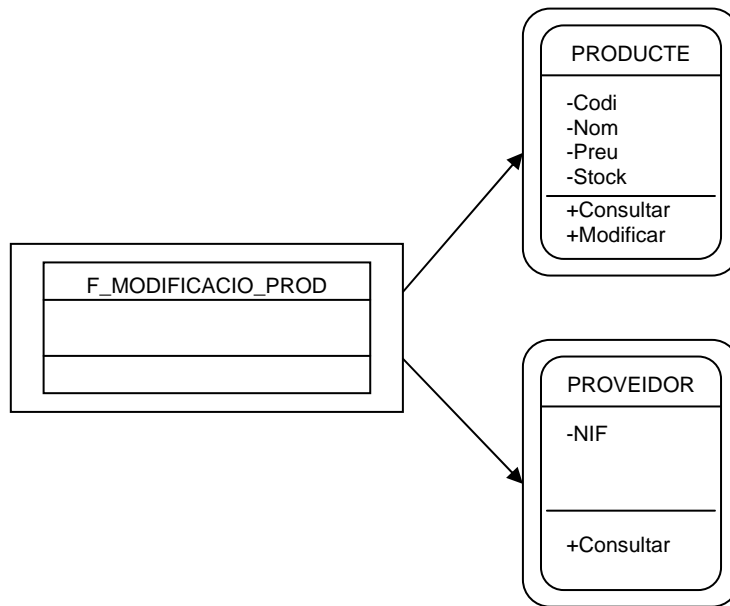
Codi:

Nom:

Preu: €

Stock:

Nom proveïdor:



F_BUSCAR_PROD:

Finestra des d'on buscarem les dades d'un producte.

BUSCAR PRODUCTE

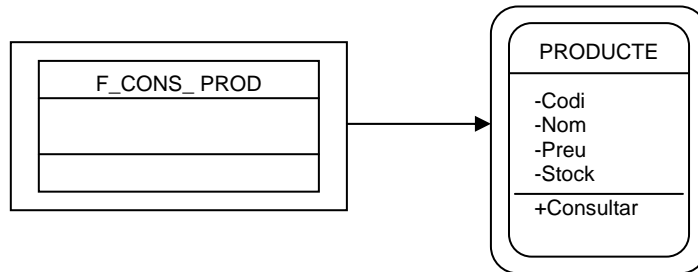
Codi:

Nom:

Preu: €

Stock:

Nom proveïdor:



F_ALTA_TRACTAMENT:

Finestra on l'usuari podrà donar d'alta els tractaments.

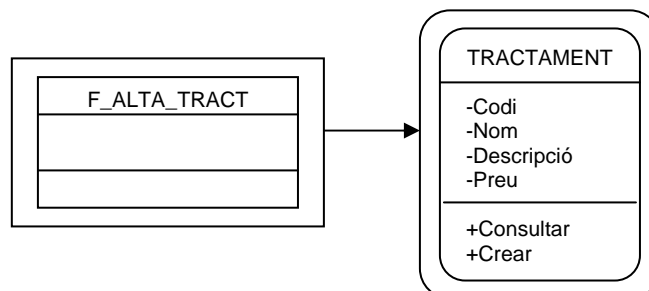
ALTA TRACTAMENT

Codi:

Nom:

Descripció:

Preu: €



F_MODIFICAR_TRACTAMENT:

Finestra a la qual s'accedirà per tal de poder modificar la informació d'un tractament.

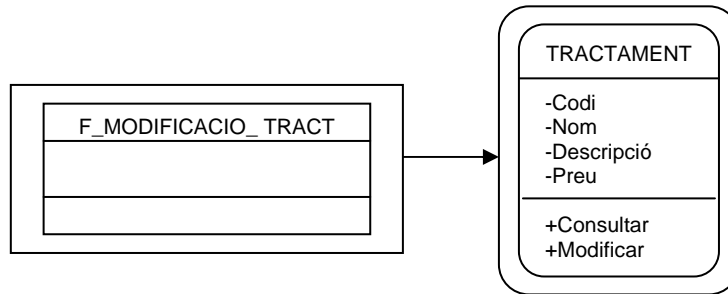
MODIFICAR TRACTAMENT

Codi:

Nom:

Descripció:

Preu: €



F_BUSCAR_TRACTAMENT:

Finestra des d'on buscarem les dades d'un tractament.

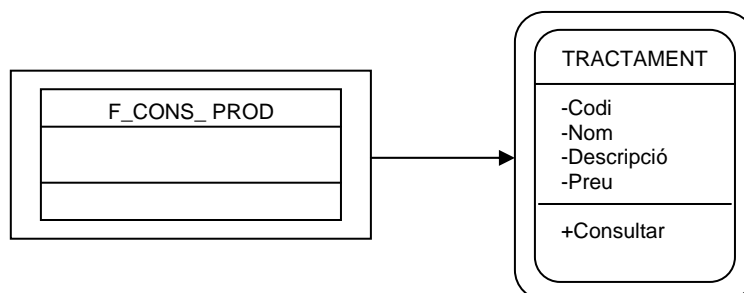
BUSCAR TRACTAMENT

Codi:

Nom:

Descripció:

Preu: €



F_ALTA_OFERTA:

Finestra que dona l'opció a posar un tractament amb oferta durant unes dates determinades.

ALTA OFERTA

Codi: 3

Nom tractament:

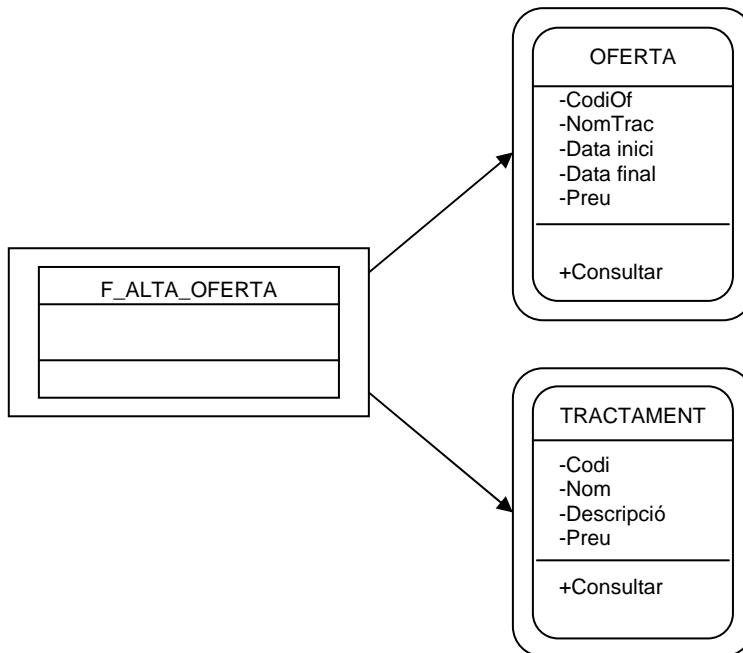
Data inici: 31/05/2007

Data final: 31/05/2007

Preu: €

Donar Alta

Sortir



F_BAIXA_OFERTA:

Finestra on accedirem per tal de donar de baixa una oferta d'un tractament.

BAIXA OFERTA

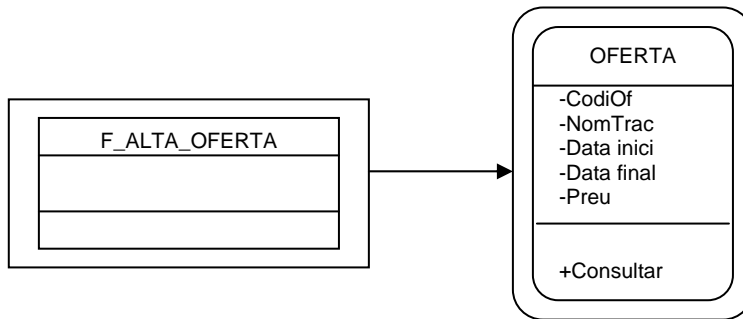
Codi:

Nom tractament:

Data inici:

Data final:

Preu: €



F_ALTA_USUARI:

Finestra des d'on es donarà d'alta un nou usuari.

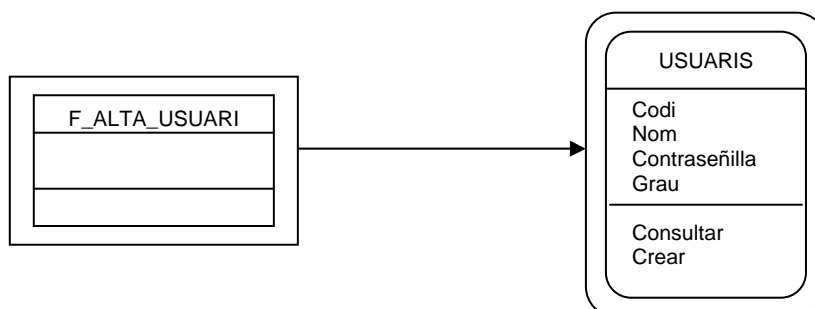
ALTA USUARI

Codi:

Usuari:

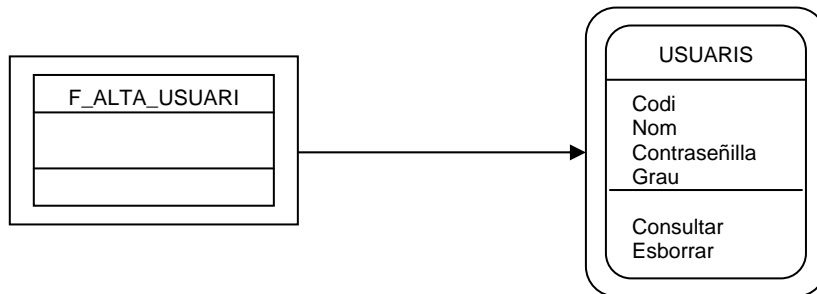
Contrassenya:

Grau:

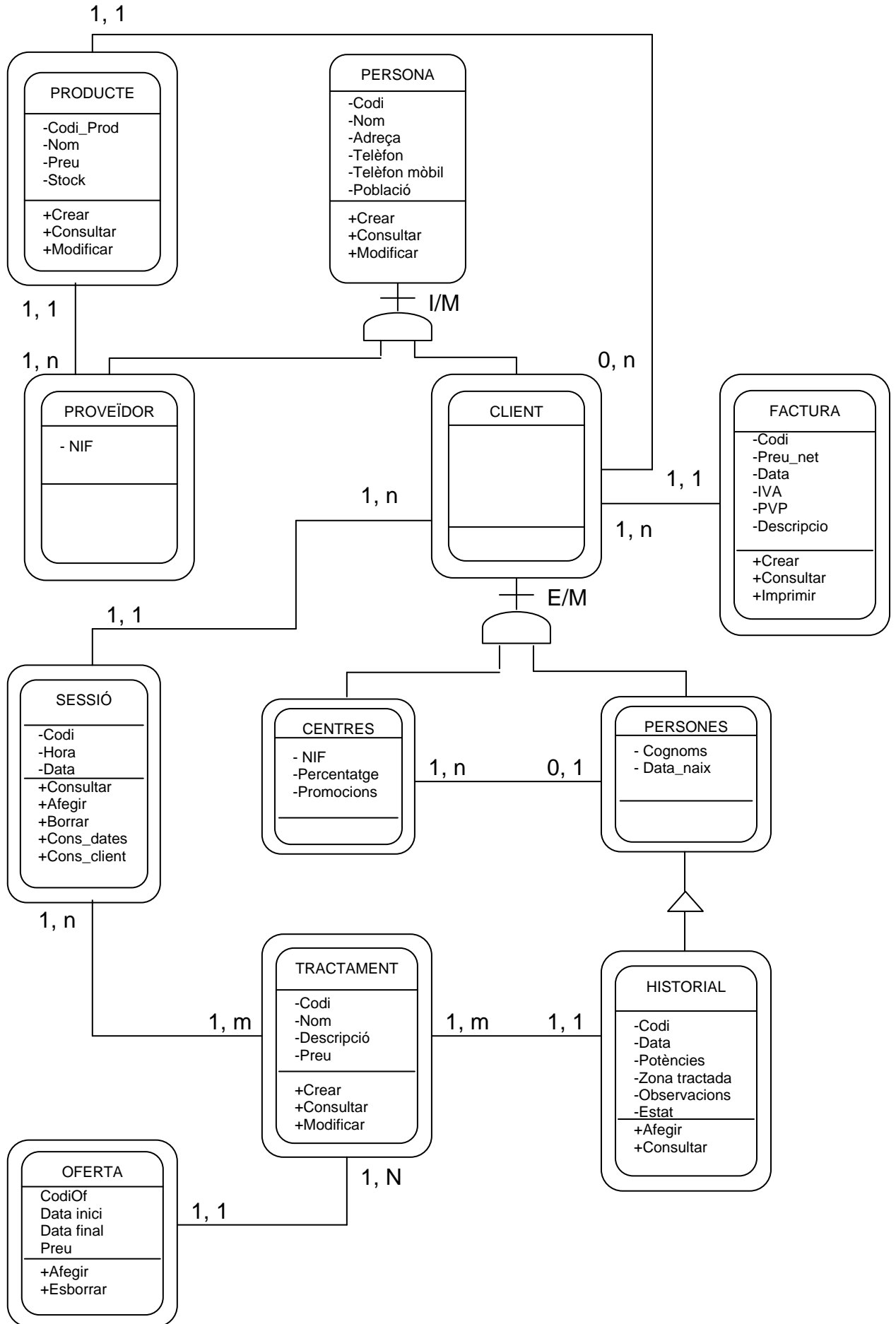


F_BAIXA_USUARI:

Finestra des d'on es donarà de baixa un usuari.



Una vegada vistes les interfícies, anem a veure el diagrama de classes ampliat, on s'observaran les noves propietats trobades, els serveis. A més observarem que davant de cada atribut, apareix el símbol "-", això significa que són atributs privats. Són atributs privats perquè només són accessibles des dels serveis de la pròpia classe. En canvi, els serveis porten el símbol "+", vol dir que són atributs públics, per tant accessibles des de qualsevol classe. És necessari que sigui així per tal de poder accedir-hi des de les classes controladores, però ha es procura en el moment de fer el disseny de continuar tenint una bona encapsulació.



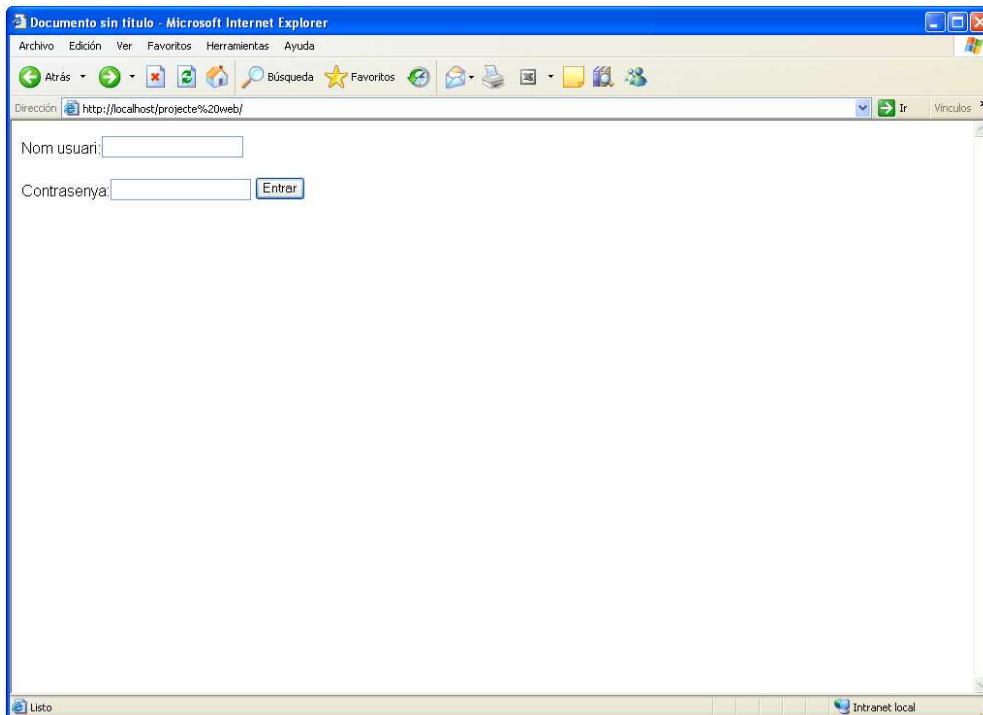
4.2.- Disseny d'interfícies en l'entorn web:

Com ja s'ha comentat anteriorment, la nostra aplicació també consta d'una part web. En la pàgina web creada, en primer lloc cal dir que és necessari un nom d'usuari i una contrasenya per tal de poder-hi entrar, la única manera de poder-te donar d'alta és que l'autònom ho faci des de l'aplicació personal. Una vegada ja connectat, depenent del grau que es tingui, per exemple grau=1 significarà que pots accedir a tots els llocs, en canvi grau=2, podràs accedir a les dades de clients, tractaments i a les agendes, on podràs consultar i modificar les dades.

Ara veurem les diferents interfícies de comunicació dissenyades per tal de poder dur a terme l'entorn web:

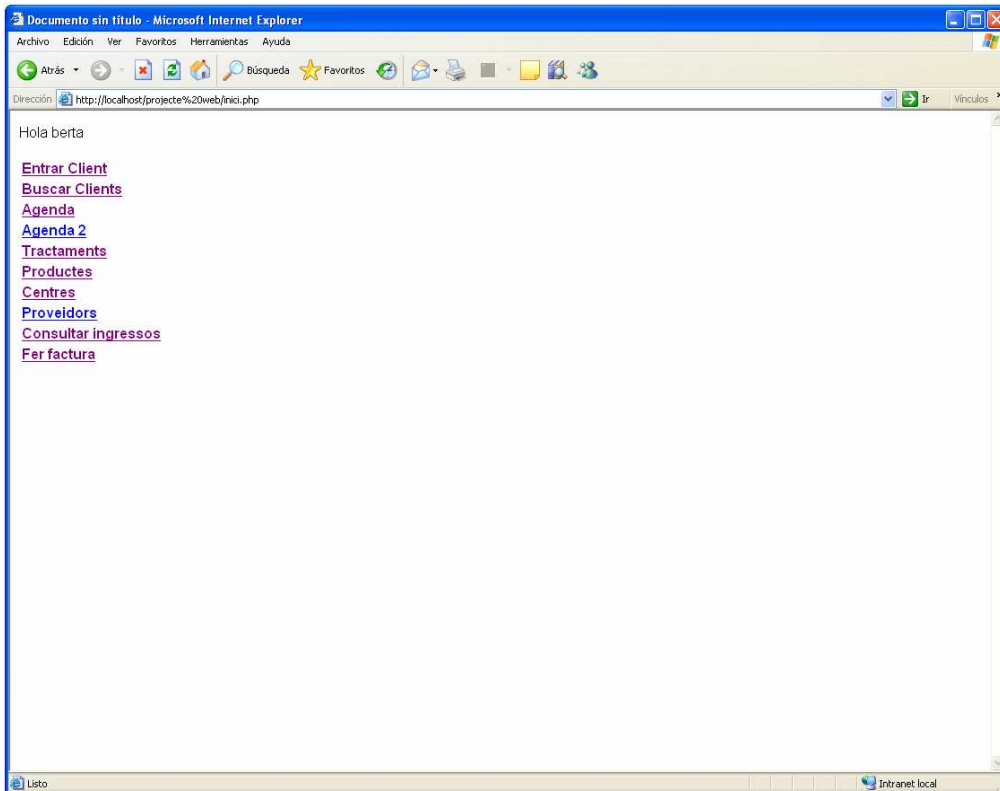
P_INICIAL:

Pàgina on hem de introduir el nom d'usuari i la contrasenya per tal de poder accedir a la pagina d'inici.



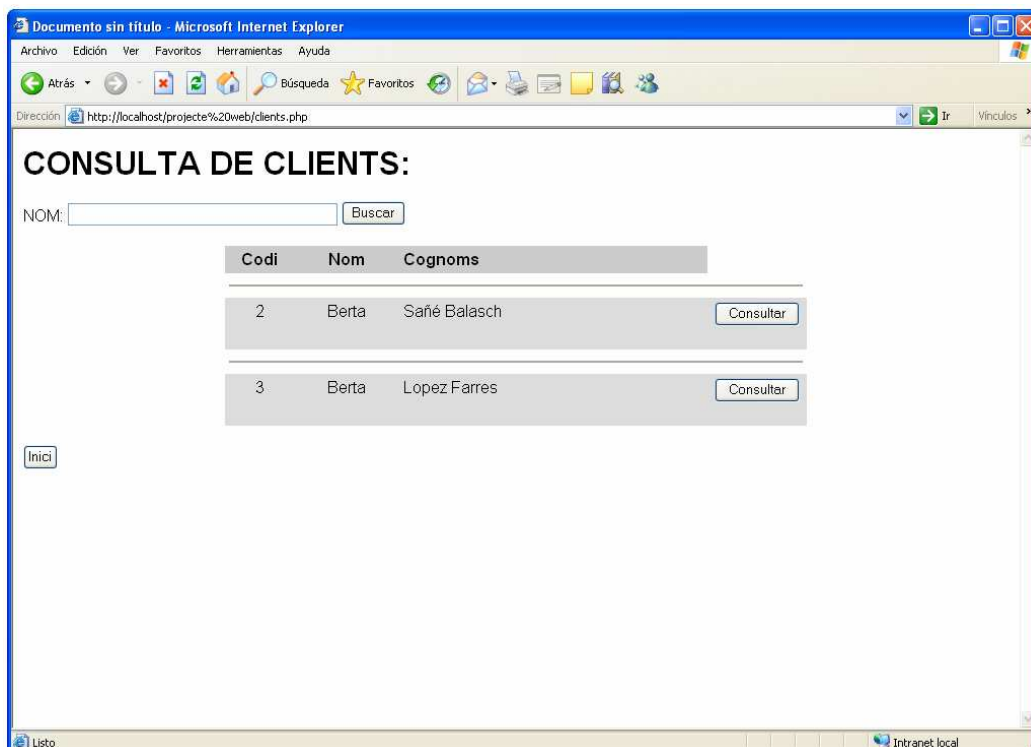
P_MENÚ_INICI:

Pàgina on es mostra el menú dels llocs on podem accedir.

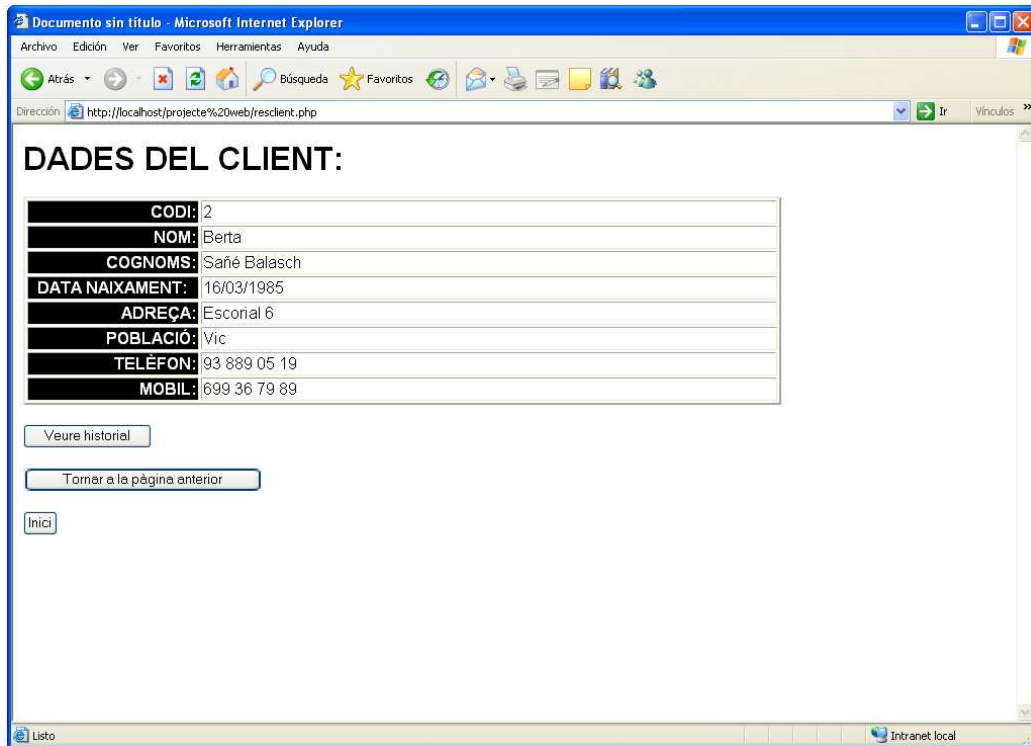


P_BUSCAR_PERS:

Pàgina on anirem per tal de buscar qualsevol informació referent a un client.

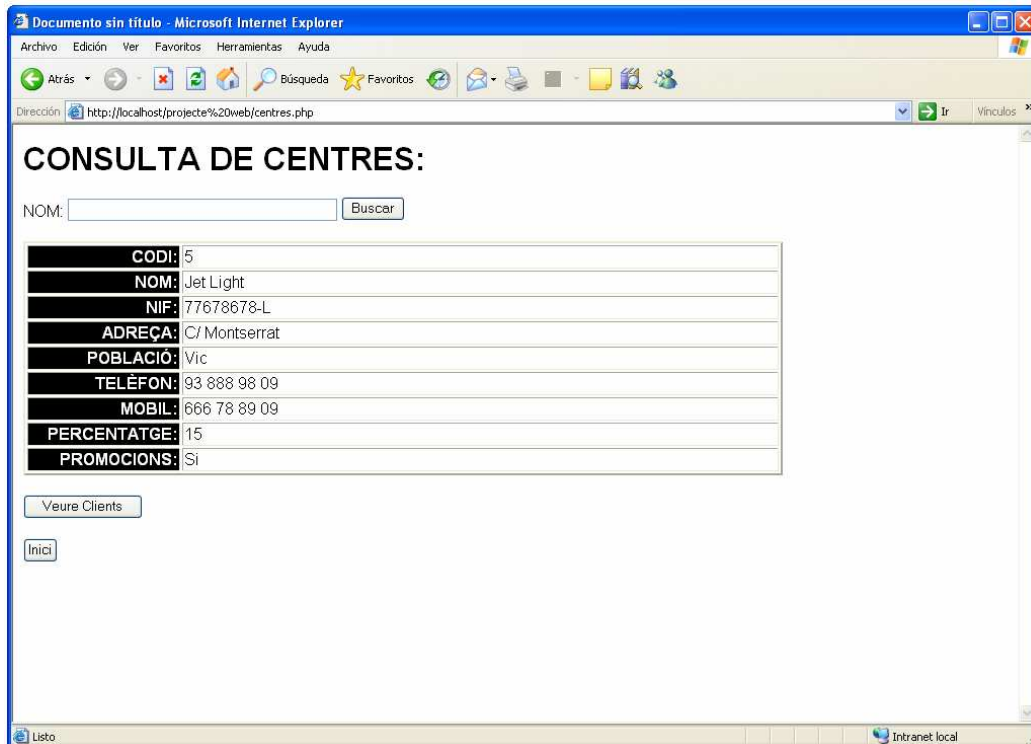


Al prémer el botó consultar se'ns obra la informació del client, des d'on podem consultar el seu historial.

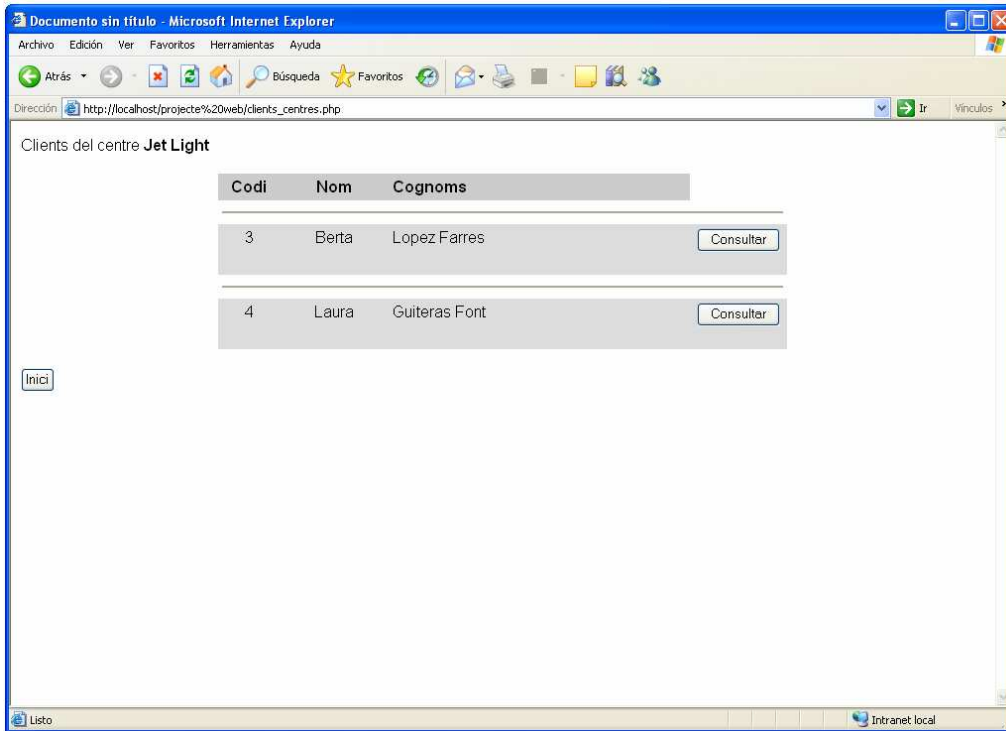


P_BUSCAR_CENTRE:

Pàgina des de la qual buscarem la informació d'un centre.

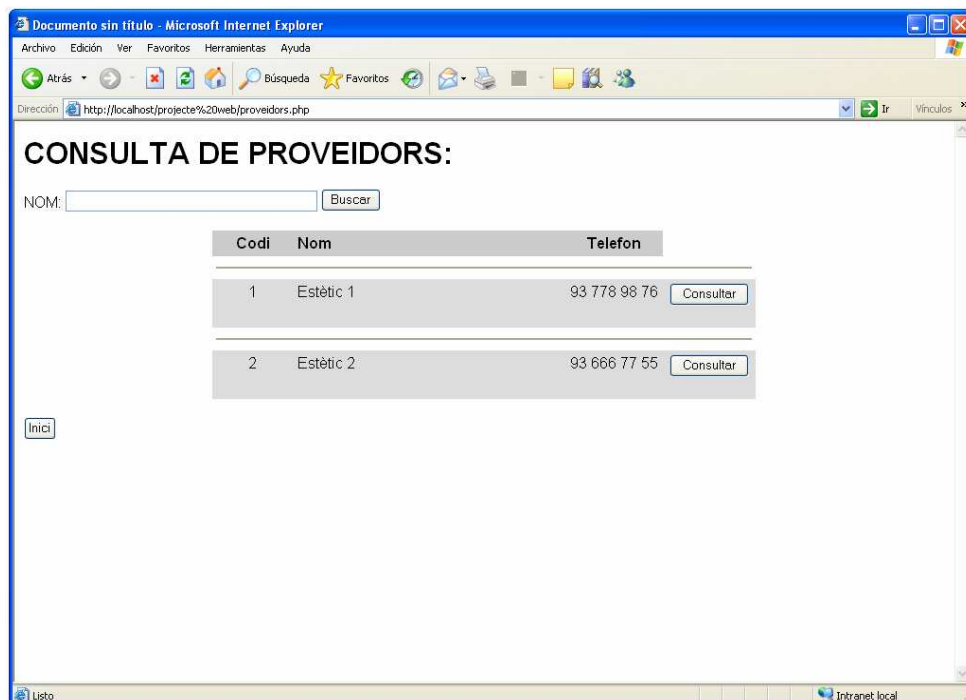


Una vegada mostrada la informació del centre, se'ns permet veure els clients que te cada centre associats, i finalment la informació d'aquests.

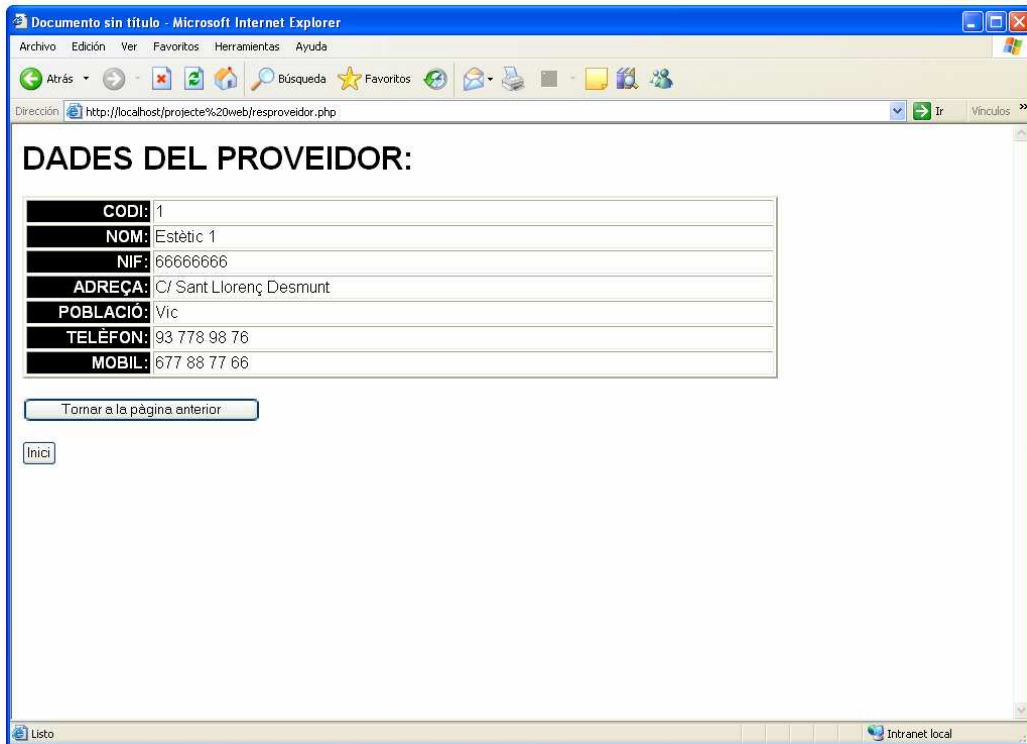


P_BUSCAR_PROV:

Pàgina que consultarem per a obtenir la informació d'un dels nostres proveïdors. En un primer moment ens sortirà una llista amb tots els proveïdors que continguin el fragment consultat, en el que mostra l'exemple 'este'.

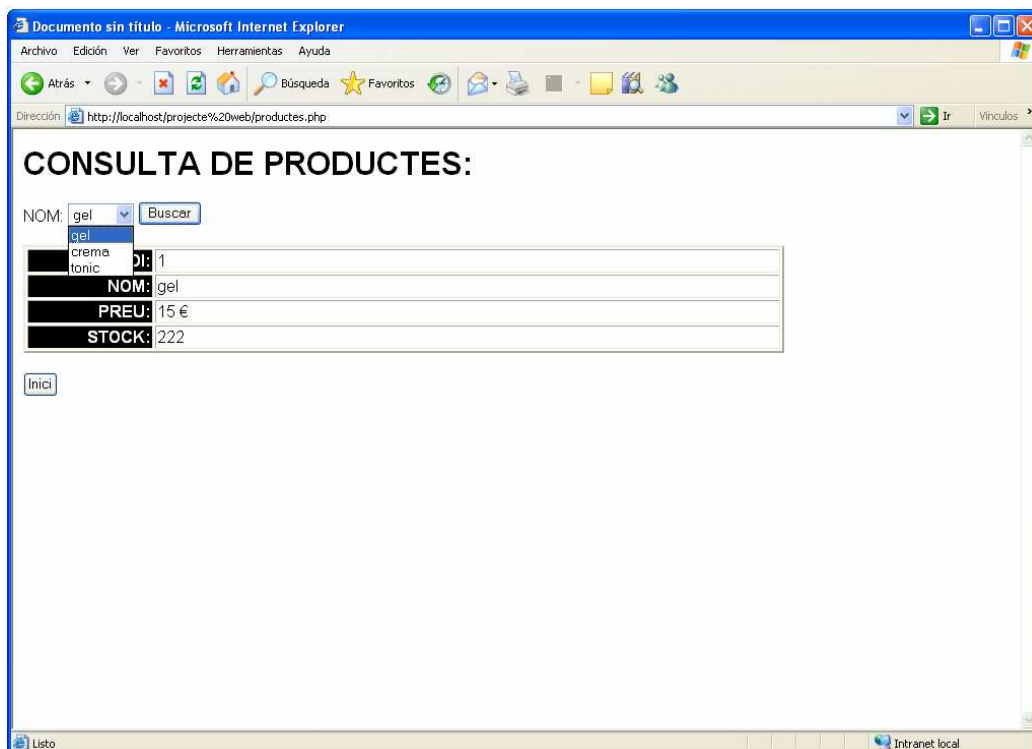


Una vegada mostrades les diferents possibilitats, es consulta el que ens interessa a nosaltres, i se'ns mostraran totes les dades entrades.



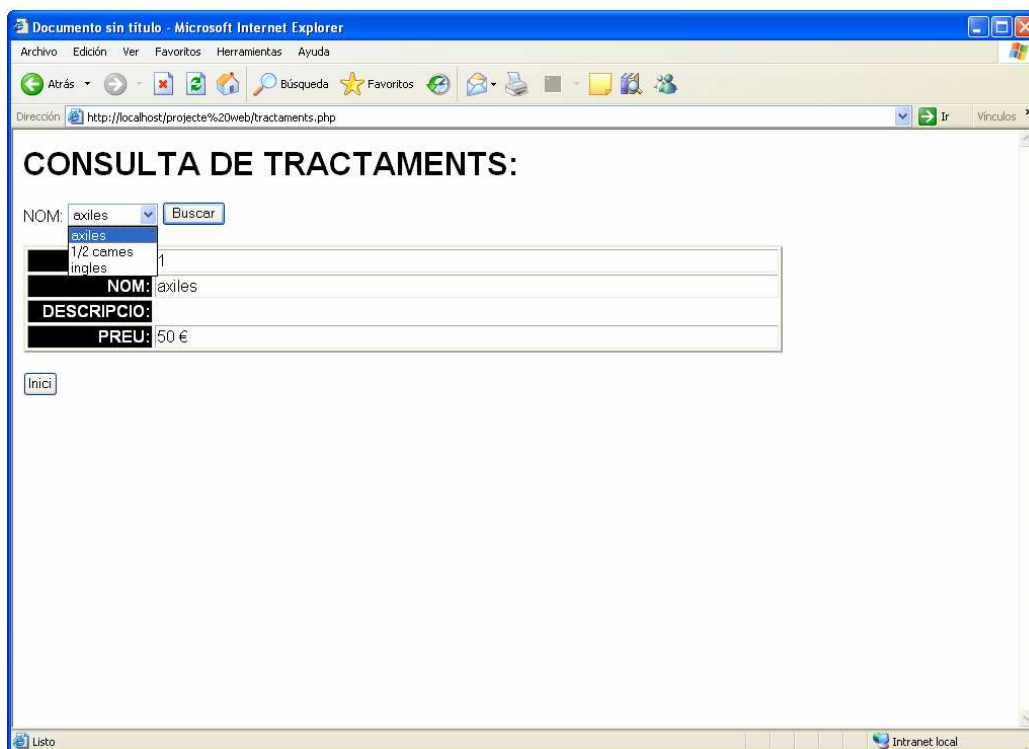
P_BUSCAR_PROD:

Pàgina a consultar per a saber el preu i les dades d'un dels productes que venem.



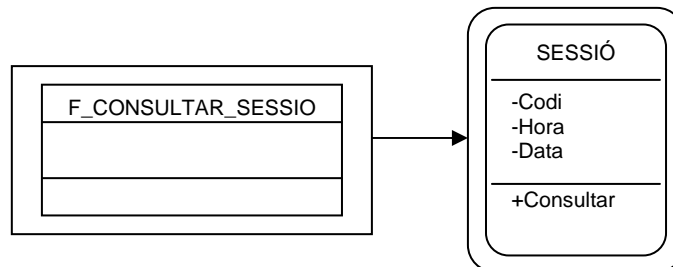
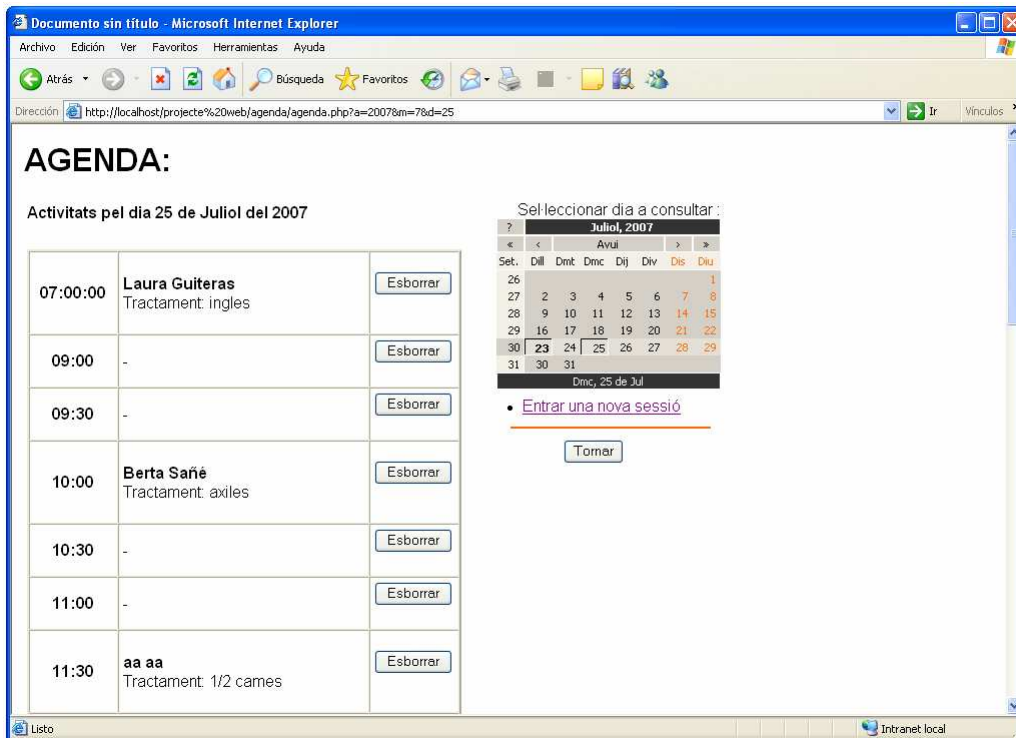
P_BUSCAR_TRACTAMENT:

Pàgina on trobarem la informació dels tractaments que oferim.



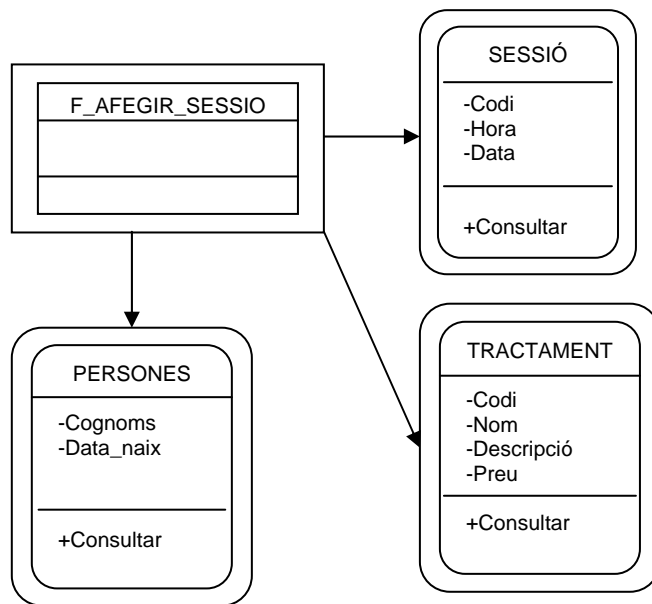
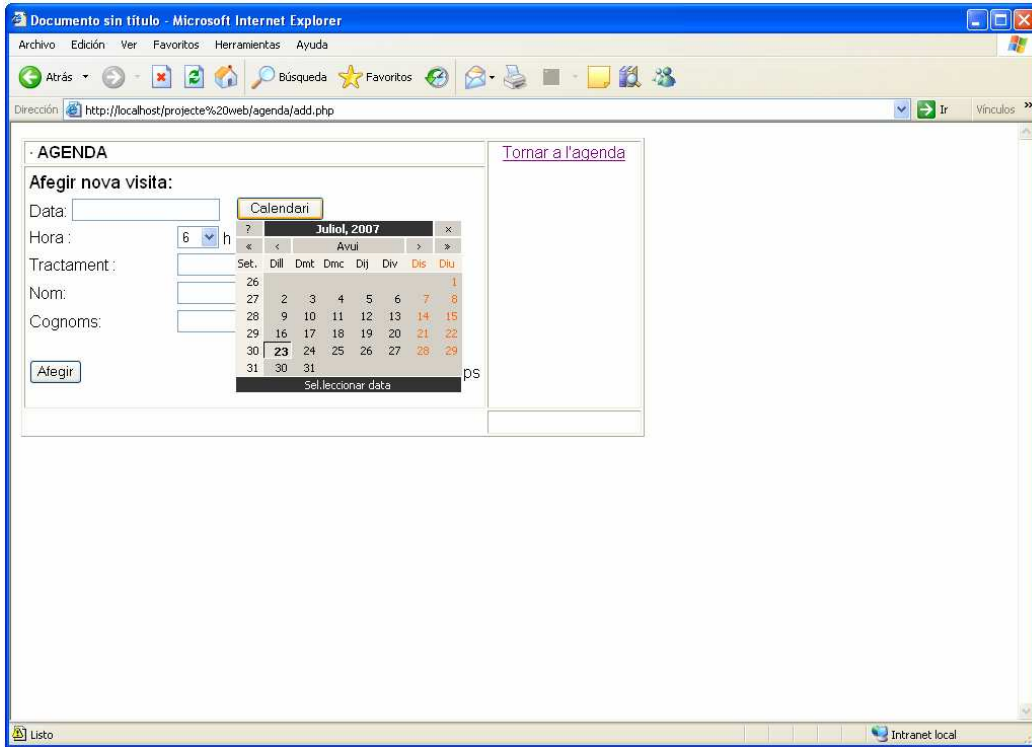
P_CONSULTAR_SESSIÓ:

Pàgina des d'on es portarà un control de les hores que tenim ocupades i lliures en un dia seleccionat, a més en qualsevol moment es podrà donar de baixa una visita.



F_AFEGIR_SESSIÓ:

Finestra a la qual accedirem alhora de voler afegir una nova visita a l'agenda.



F_AFEQIR_HISTORIAL:

Finestra a la qual anirem per tal d'ampliar l'historial d'un client, on posarem les dades que més interessants i ho adjuntarem a l'historial.

AFEGIR DADES NOVES A L'HISTORIAL:

Codi:

Data:

Zona tractada: (dropdown menu open with options: axiles, 1/2 cames, ingles)

Potències:

Observacions:

Estat del tractament: (dropdown menu)

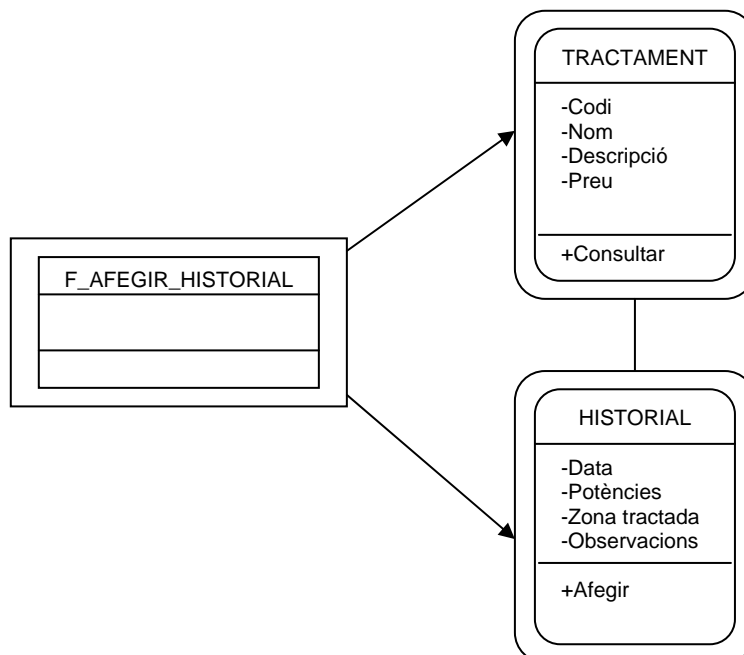
PRODUCTE COMPRAT:

Codi:

Data:

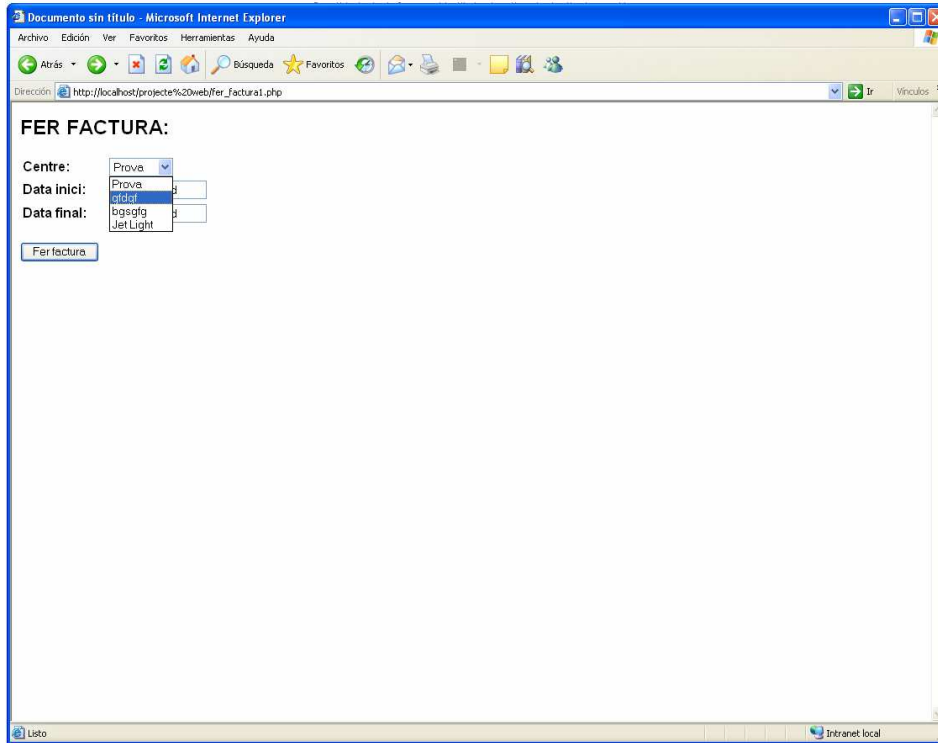
Producte: (dropdown menu)

Quantitat:

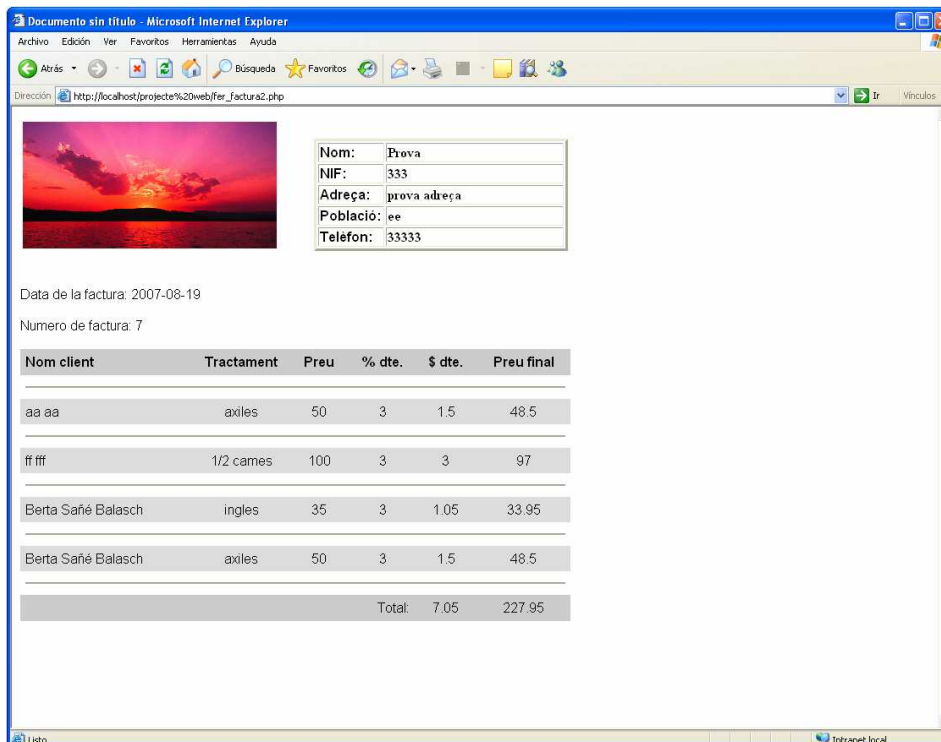


F_FER_FACTURA:

Finestra on es realitzaran les factures per als centres associats. En aquesta primera finestra, s'haurà d'escollir el centre en el qual vols fer la factura, i les dates de inici i fi del temps que vols facturar.



Una vegada omplertes les dades necessàries, es prem el botó “Fer factura”.



Resultat de la factura una vegada impresa:

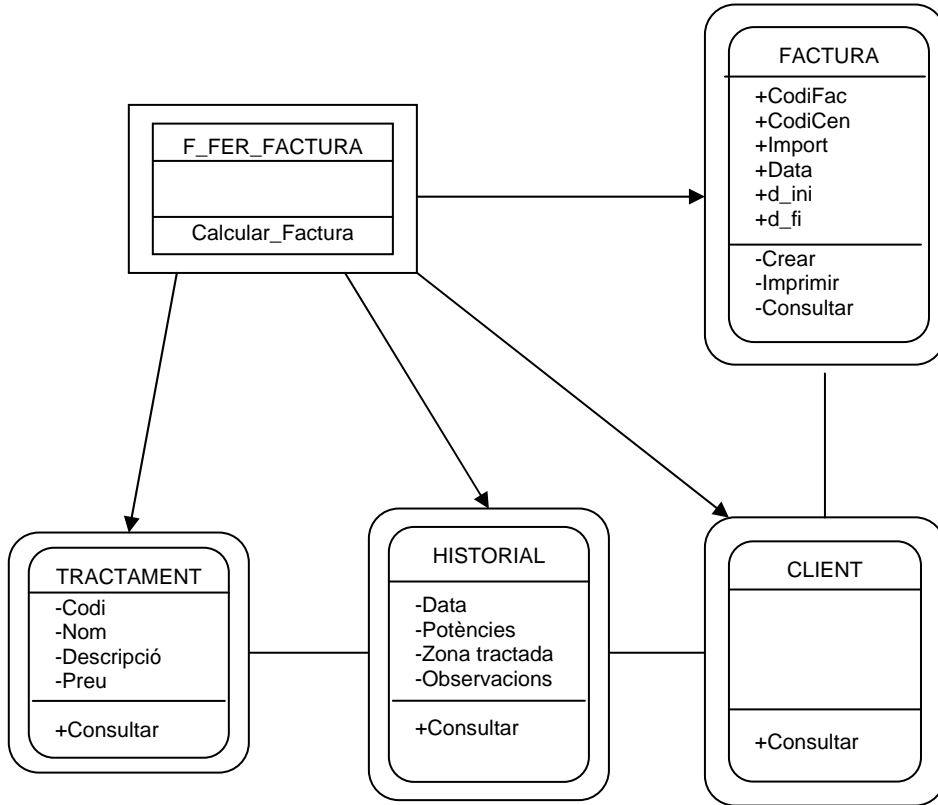


Nom:	Prova
NIF:	333
Adreça:	prova adreça
Població:	ee
Telèfon:	33333

Data de la factura: 2007-08-19

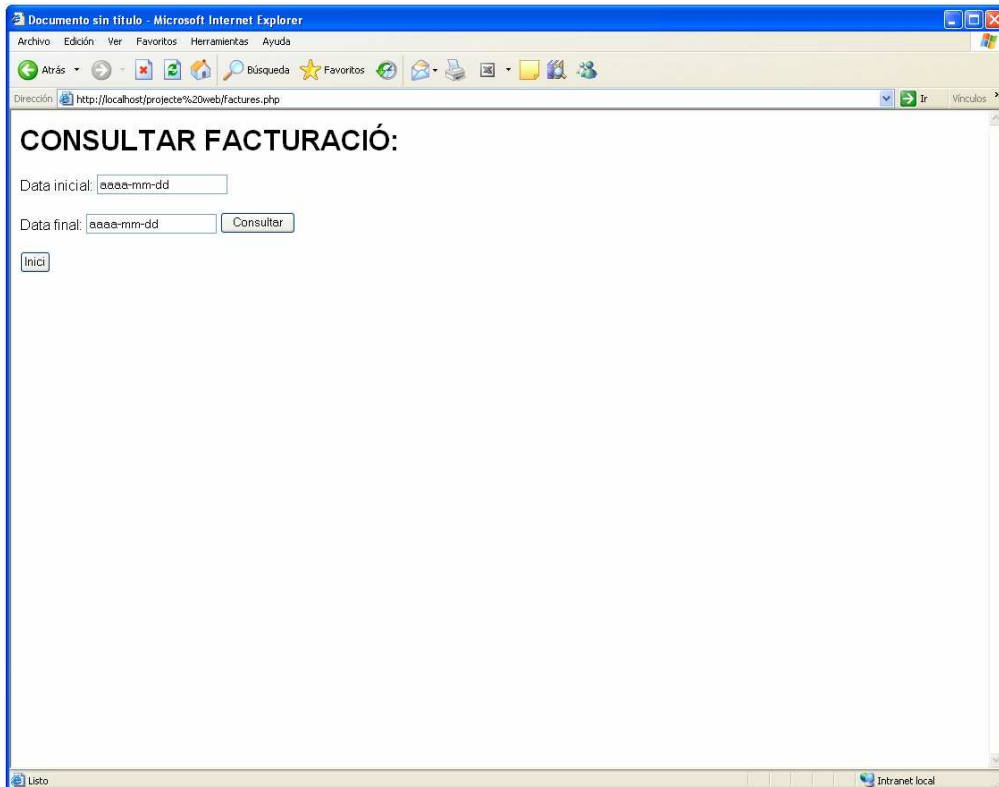
Numero de factura: 7

Nom client	Tractament	Preu	% dte.	\$ dte.	Preu final
aa aa	axiles	50	3	1.5	48.5
ff fff	1/2 cames	100	3	3	97
Berta Sañé Balasch	ingles	35	3	1.05	33.95
Berta Sañé Balasch	axiles	50	3	1.5	48.5
Total:				7.05	227.95

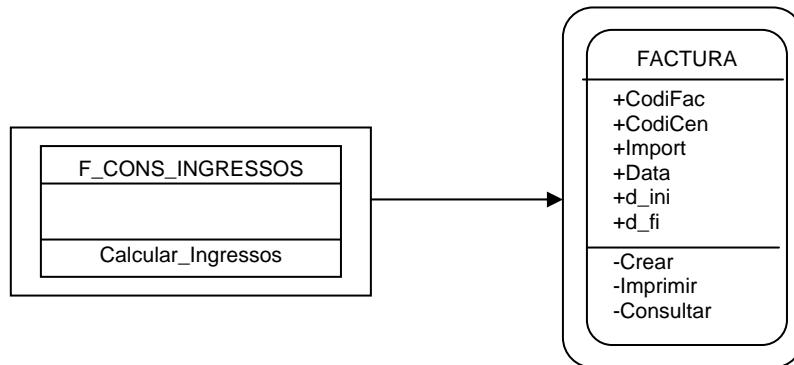
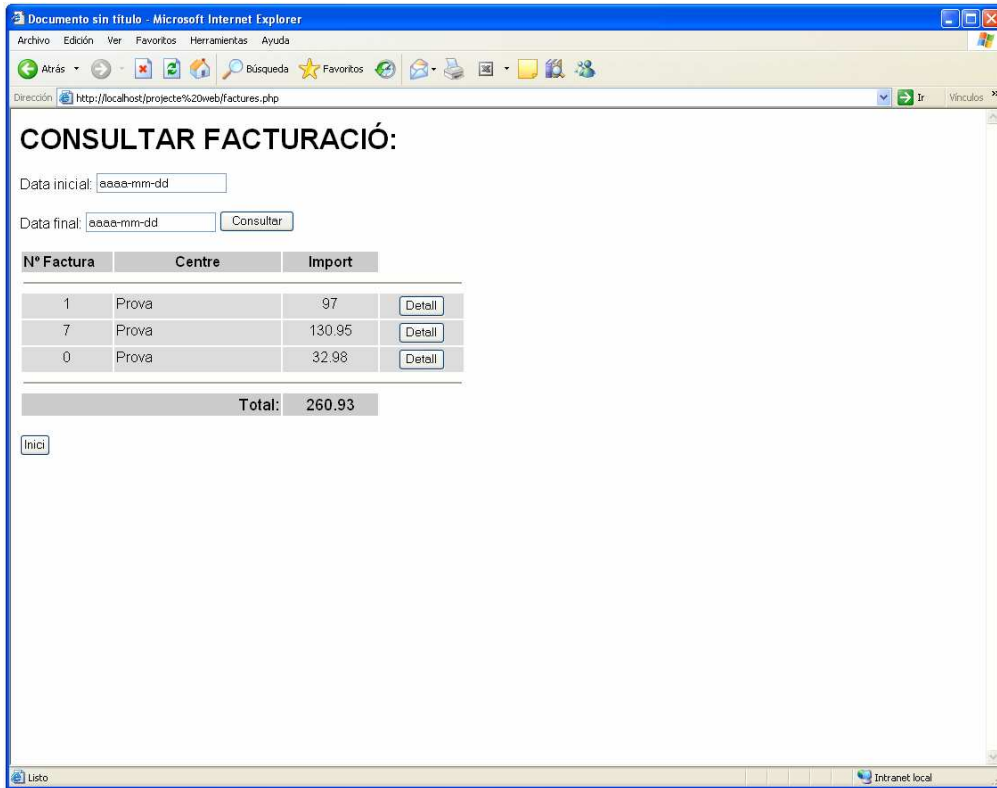


F_CONSULTAR_INGRESSOS:

Finestra on es mostraran els ingressos que hem tingut en un període de temps especificat per a l'usuari.



Una vegada entrades les dates, es prem consultar i et surt un llistat de les factures que has fet entre les dates entrades, on s'especifica el numero de factura, el centra i l'import. Finalment pots veure el detall de la factura que t'interessi.



4.3.- Disseny de la base de dades:

Per la realització de l'aplicació hem escollit com a sistema gestor de bases de dades (SGBBDD) el MySQL server, tot i que el disseny que hem realitzat és orientat a objectes, i aquest SGBBDD no ho és. La raó d'aquesta elecció es que necessitàvem un software barat al mateix temps que efectiu per a les nostres necessitats. A més al tenir l'aplicació web, és necessari tenir un servidor apache i el php, i varem trobar un software que ja incorporava les tres coses, el php, l'apache i el MySQL.

Per tal de fer la traducció del diagrama de classes a un model relacional, hem seguit els següents passos:

- 1er pas: cada classe passa a ser una relació. D'aquesta manera obtenim les següents relacions: historial, usuari, producte, proveïdors, tractaments, factures, ofertes i agenda.
- 2on pas: depenent del tipus de la interrelació que tinguem (1:1, 1:N, N:M) fem una traducció o una altre.

En el cas de una interrelació 1:1, agafem la clau primària d'una taula i la posem com a clau forana de l'altre. Això es pot observar en el següent exemple:

```
create table tractaments(Codi int(3), Nom char(15) PRIMARY KEY,
    Descripcio text(500), Preu int(3));
create table ofertes(Codiof int(3), Data_inici date, Data_fi date, Preu
    int(3), NomTrac char(15), PRIMARY KEY (Codiof, Data_inici,
    Data_fi), FOREIGN KEY (NomTrac) REFERENCES
    tractaments(Nom));
```

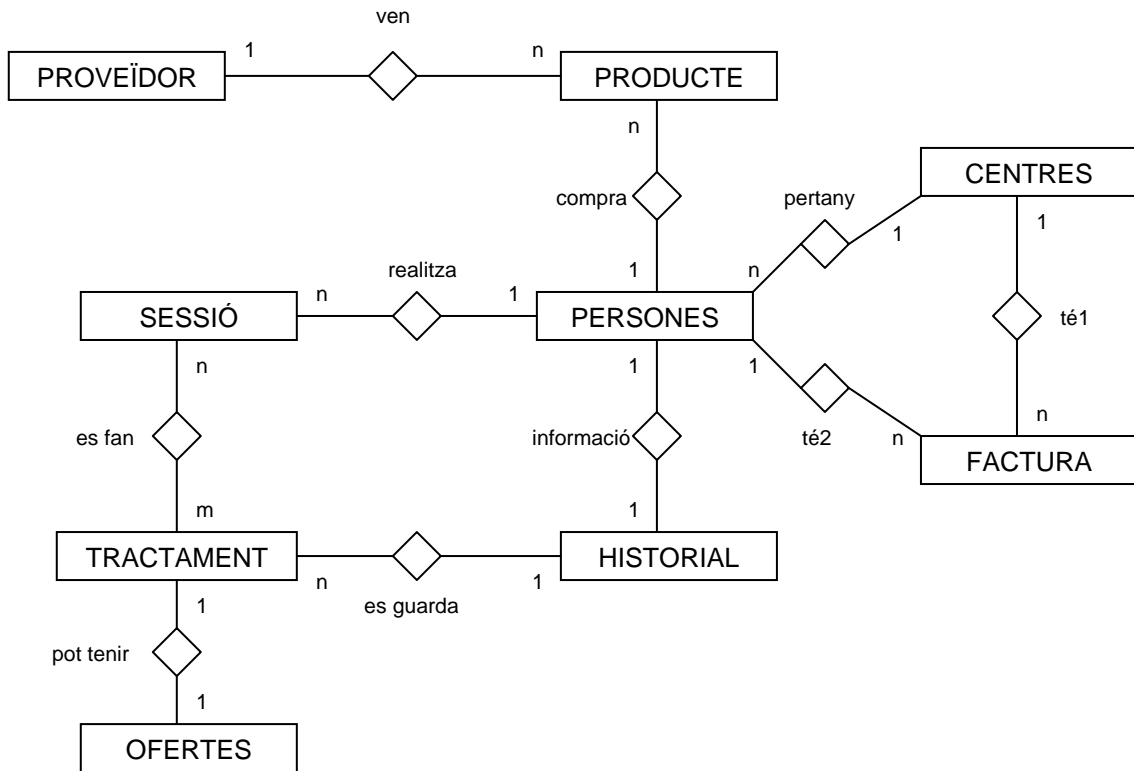
En el cas de ser una interrelació 1:N, la clau primària de la relació que té 1, passa a ser clau forana al cantó de la N, mirem-ne un exemple:

```
create table proveidors(Codi int(3) PRIMARY KEY, Nom char(10),
    Telefon char(15), Mobil char(15), Adreça char(25), Poblacio
    char(15), NIF char(9));
create table productes(Codi int(3) PRIMARY KEY, Nom char(15),
    CodiProv int(3), Preu char(6), Stock int(3), FOREIGN KEY
    (CodiProv) REFERENCES proveidors(Codi));
```

Finalment anem a veure l'últim cas que ens podem trobar, que la interrelació sigui N:M. En aquest cas es crea una nova relació amb com a mínim, les 2 claus primàries.

- 3er pas: per últim tenim la part de les generalitzacions, nosaltres tenim la generalització de persona, on tenim proveïdors i clients, i la generalització de clients on podem trobar centres i persones. Per tal de fer aquesta traducció, hem agafat únicament els nodes de l'arbre, és a dir hem creat les taules de Proveïdors, Centres i Persones.

Tot seguit anem a veure el model relacional que ens queda una vegada traduït el diagrama de classes:



4.3.1.- Descripció de les taules:

A continuació es farà una descripció del contingut de les taules, l'estructura que s'utilitza serà: en primer lloc es posarà el nom de la taula, i a continuació es posaran tots els atributs de les taules, descrivint el tipus de dades que és i si es dona el cas informar de si es tracta d'una clau primària o forana.

CENTRES:

Codi: Numèric – Clau primària

NIF: Numèric

Nom: Alfabètic

Adreça: Alfabètic

Telèfon: Alfabètic

Mòbil: Alfabètic

Població: Alfabètic

Percentatge: Numèric

Promocions: Booleà

PERSONES:

Codi: Numèric – Clau primària

Nom: Alfabètic

Cognoms: Alfabètic

Data naix: Date

Adreça: Alfabètic

Telèfon: Alfabètic

Mòbil: Alfabètic

Població: Alfabètic

Nom Centre: Numèric – Clau forana de centres

PROVEÏDOR:

Codi: Numèric – Clau primària

NIF: Numèric

Nom: Alfabètic

Adreça: Alfabètic

Telèfon: Alfabètic

Mòbil: Alfabètic

Població: Alfabètic

PRODUCTE:

Codi Prod: Numèric – Clau primària

Codi Prov: Numèric – Clau forana de proveïdors

Nom: Alfabètic

Preu: Numèric

Stock: Numèric

TRACTAMENT:

Codi: Numèric – Clau primària

Nom: Alfabètic

Descripció: Alfabètic

Preu: Numèric

FACTURA:

CodiFact: Numèric – Clau primària

Codi Centre: Numèric – Clau forana de centres

Data: Date

Import: Numèric

D_inici: Date

D_fi: Date

HISTORIAL:

Codi: Numèric – Clau forana de persones – Clau primària

Data: Date – Clau primària

Potències: Numèric

Zona tractada: Alfabètic – Clau primària

Observacions: Alfabètic

Estat: Alfabètic

OFERTA:

CodiOf: Numèric – Clau primària

Data_inici: Date – Clau primària

Data_final: Date – Clau primària

Preu: Numèric

Codi Tract: Numèric – Clau forana de tractaments i clau primària

SESSIO:

Codi: Numèric – Clau primària

Data: Date – Clau primària

Hora: Time – Clau primària

Tractament: Alfabètic – Clau forana de tractaments

Nom_cli: Alfabètic – Clau forana de persones

Cog_cli: Alfabètic – Clau forana de persones

USUARIS:

Codi: Numèric – Clau primària

Nom: Alfabètic

Contrasenya: Alfabètic

Grau: Numèric

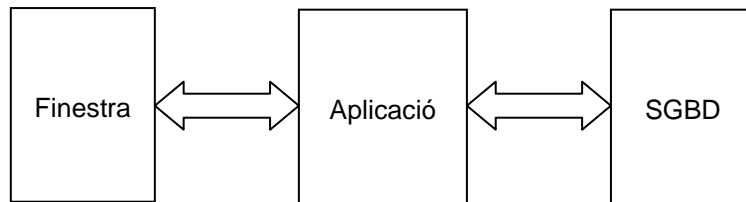
Una vegada vista la descripció de les taules, cal dir que hi ha alguns inconvenients, coma ara la integritat de les dades. Què volem dir amb aixó, doncs simplement fer el lligam que hi ha entre les taules, el SGBD que hem escollit, no ho fa per ell mateix, la qual cosa vol dir que ho hem tingut de fer tot nosaltres en el moment de fer la implementació dels programes.

4.4.- Disseny de programes:

4.4.1.- Disseny en l'entorn personal:

Per tal de portar a terme l'implementació de l'entorn personal hem utilitzat el *Borland C++ Builder*, ja que ens era un software conegut.

Anem a recordar els components software utilitzats en l'entorn personal. Aquest ha estat dividit en tres parts: finestra, aplicació i finalment la base de dades.



1.- FINESTRA:

En la classe finestra implementem amb objectes, tenim dos tipus de propietats, els atributs, i els serveis. En els atributs i trobem diferents camps, com és el codi, el nom, l'stock, el preu i el nom del proveïdor en el cas de l'exemple que mostrarem a continuació. Pel que fa als serveis, aquests estan associats els events del programa, que en el nostre exemple seria el cas de donar una alta o sortir de la finestra. Utilitzem a part de l'interfície de comunicació per a fer una bona encapsulació. Fem les crides dels serveis dels objectes i així aquests no s'han de conèixer.

Seguidament mostrarem el codi utilitzat per tal de fer una alta d'un producte:

```

#include <StdCtrls.hpp>
#include <Forms.hpp>
#include "Unit3.h" //connexió amb la base de dades
//-----
class TF_ALTA_PROD : public TForm
{
__published: // IDE-managed Components
    TLabel *Label1;
    TEdit *CODI;
    TEdit *NOM;
    TEdit *PREU;
    TEdit *STOCK;
    TLabel *Label2;
    TLabel *Label3;
    TLabel *Label4;
    TLabel *Label5;
    TButton *Alta;
  
```

```

        TButton *Sortir;
        TLabel *Label6;
        TEdit *NOM_PROV;
        TLabel *Label7;
        void __fastcall AltaClick(TObject *Sender);
        void __fastcall SortirClick(TObject *Sender);
private:    // User declarations
public:    // User declarations
        __fastcall TF_ALTA_PROD(TComponent* Owner);
};
//-----
extern PACKAGE TF_ALTA_PROD *F_ALTA_PROD;
//-----
#endif

```

Ara anem a veure la part del cpp:

```

#include <vcl.h>
#pragma hdrstop
#include "Unit9.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TF_ALTA_PROD *F_ALTA_PROD;
//-----
__fastcall TF_ALTA_PROD::TF_ALTA_PROD(TComponent* Owner)
        : TForm(Owner)
{
}
//-----

void __fastcall TF_ALTA_PROD::AltaClick(TObject *Sender)
{
int codip;
//Comprovem que hi hagi unes dades mínimes entrades
if ((NOM->Text=="") | (PREU->Text=="") | (NOM_PROV->Text==""))
{
        Application->MessageBoxA("Falten dades del producte",          "Entrada
        incorrecta",MB_OK|MB_ICONASTERISK);
}
else
{
        //Mirem que no hi hagi cap altre producte amb el mateix nom
TLocateOptions Opts;
Opts.Clear();
Opts << loPartialKey;
BD->productes->Open();
if(BD->productes->Locate("Nom",F_ALTA_PROD->NOM->Text,Opts))
{
        Application->MessageBoxA("Producte          amb          nom          duplicat",
        "ATENCIO",MB_OK|MB_ICONWARNING);
}
}
}

```

```

else
{
    //Comprovem que el proveïdor existeixi a la base de dades
    TLocateOptions Opts;
    Opts.Clear();
    Opts << loPartialKey;
    BD->proveidors->Open();
    if(!BD->proveidors->Locate("Nom",F_ALTA_PROD->NOM_PROV->
    Text,Opts))
    {
        Application->MessageBoxA("Proveïdor inexistent a la
        base de dades. Primer has d'entrar les seves
        dades.", "Nom inexistent", MB_OK|MB_ICONWARNING);
        F_ALTA_PROD->CODI->Clear();
        F_ALTA_PROD->NOM->Clear();
        F_ALTA_PROD->PREU->Clear();
        F_ALTA_PROD->STOCK->Clear();
        F_ALTA_PROD->NOM_PROV->Clear();
        BD->proveidors->Close();
        Close();
    }
    else
    {
        //Entrem les dades a la base de dades
        codip=BD->proveidors->FieldByName("Codi")->AsInteger;
        BD->productes->Open();
        BD->productes->Last();
        BD->productes->Insert();
        BD->productes->FieldByName("Codi")->AsInteger=
            StrToInt(F_ALTA_PROD->CODI->Text);
        BD->productes->FieldByName("Nom")->AsString=
            F_ALTA_PROD->NOM->Text;
        BD->productes->FieldByName("Preu")->AsString=
            F_ALTA_PROD->PREU->Text;
        BD->productes->FieldByName("Stock")->AsInteger=
            StrToInt(F_ALTA_PROD->STOCK->Text);
        BD->productes->FieldByName("CodiProv")->AsInteger=
            codip;
        BD->productes->Post();
        BD->productes->Close();
        Application->MessageBoxA("Producte introduït a la
        base de dades", "Entrada correcta",
            MB_OK|MB_ICONASTERISK);
        F_ALTA_PROD->CODI->Clear();
        F_ALTA_PROD->NOM->Clear();
        F_ALTA_PROD->PREU->Clear();
        F_ALTA_PROD->STOCK->Clear();
        F_ALTA_PROD->NOM_PROV->Clear();
        BD->proveidors->Close();
        Close();
    }
}
}
}

```

```
}  
//-----  
void __fastcall TF_ALTA_PROD::SortirClick(TObject *Sender)  
{  
    Close();  
}  
//-----
```

2.- APLICACIÓ:

En aquesta part implementem els objectes propis de l'aplicació. El que aconseguim amb els objectes és que aquets heretin d'altres. Com a classes objectes podem trobar: centre, particular i proveïdor. Tot seguit podem observar el codi utilitzat per crear l'objecte centre:

```
#ifndef Unit8H  
#define Unit8H  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <SysUtils.hpp>  
#include <Db.hpp>  
#include <Dbtables.hpp>  
#include "Unit3.h" //connexió amb la base de dades  
#include "Unit4.h" //inclou classe client, que hereta els seus atributs  
  
class centre: public client{  
public:  
    AnsiString nif;  
    int codi;  
    bool buscar (AnsiString);  
    bool buscarNom (AnsiString);  
    void alta();  
};  
extern PACKAGE centre *ocent;  
#endif
```

Ara anem a veure la part del cpp:

```
#include <vcl.h>
#pragma hdrstop
#include "Unit8.h"
#pragma package(smart_init)
//-----
centre *ocent;
bool centre::buscar (AnsiString n){
    bool r;
    TLocateOptions Opts;
    BD->centres->Open();
    Opts.Clear();
    Opts << loPartialKey;
    if (BD->centres->Locate("NIF", n, Opts)){
        r=true;
        return r;
    }else{
        r=false;
        return r;
    }
    BD->centres->Close();
};
//-----
bool centre::buscarNom (AnsiString n){
    bool r;
    TLocateOptions Opts;
    BD->centres->Open();
    Opts.Clear();
    Opts << loPartialKey;
    if (BD->centres->Locate("Nom", n, Opts)){
        r=true;
        return r;
    }else{
        r=false;
        return r;
    }
    BD->centres->Close();
};
//-----
void centre::alta(){
    client::alta(BD->centres);
    BD->centres->FieldByName("Codi")->AsInteger=ocent->codi;
    BD->centres->FieldByName("Nom")->AsString=ocent->nom;
    BD->centres->FieldByName("NIF")->AsString=ocent->nif;
    BD->centres->FieldByName("Adreça")->AsString=ocent->adreça;
    BD->centres->FieldByName("Poblacio")->AsString=ocent->poblacio;
    BD->centres->FieldByName("Telefon")->AsString=ocent->telefon;
    BD->centres->FieldByName("Mobil")->AsString=ocent->mobil;
    BD->centres->Post();
    BD->centres->Close();
}
```

3.- BASE DE DADES:

Per altra banda, en la part de l'aplicació, pel que fa referència a l'accés a la base de dades, utilitzem dos maneres per tal d'accedir a ella.

La primera opció, l'utilitzem per tal de fer consultes simples a la base de dades, bàsicament les utilitzem quan ens referim a un sol registre. Per dur a terme la consulta, utilitzem les següents operacions d'accés, pròpies del C++:

- Open: Obrir la taula.
- Locate: Per localitzar un registre concret d'una taula
- Last: Per situar-nos a l'últim registre de la taula. Útil tan per donar l'últim codi més un al donar d'alta un objecte com per inserir un registre.
- Insert: Marcar que estem en mode inserir.
- Edit: Marcar que estem en mode editar.
- Delete: Esborrar el registre en curs (seleccionat amb l'operació *Locate*).
- Post: Gravar el registre en curs.
- Close: Tancar la taula.

A continuació mostrarem un petit exemple de codi de l'aplicació, que mostra la manera de connectar que s'ha explicat:

```
TLocateOptions Opts;
Opts.Clear();
Opts << loPartialKey;
BD->proveidors->Open();
if(!BD->proveidors->Locate("Nom", F_CONS_PROV->NOM->Text, Opts))
...
BD->proveidors->Close();
```

La segona manera que tenim, és utilitzant sentències SQL. Per tal de poder realitzar les consultes SQL de l'aplicació, hem creat un objecte de tipus Query (anomenat 'consulta') al DataModule. Anem a veure com realitzar una consulta mitjançant un exemple:

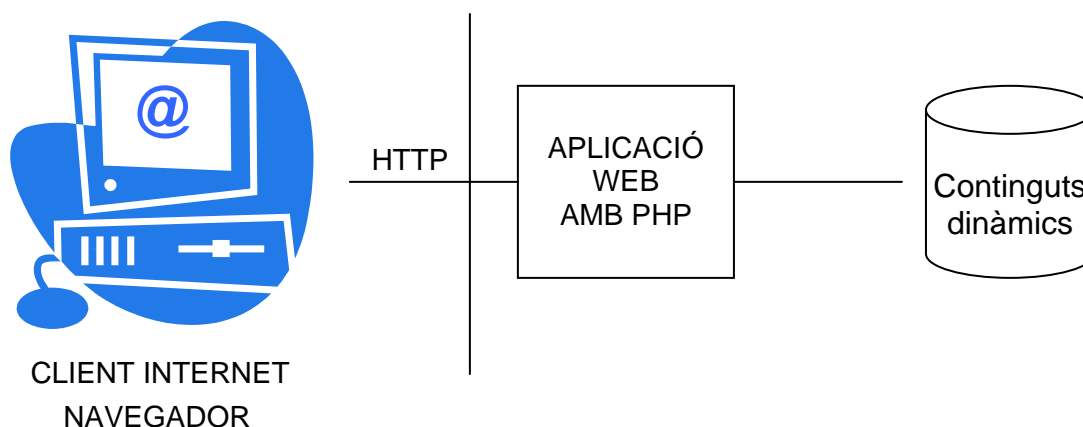
```
BD->consulta->Active=False;
BD->consulta->Close();
BD->consulta->SQL->Clear();
BD->consulta->SQL->Add("SELECT Codi, Nom, Cognoms FROM persones WHERE
Nom=:T");
BD->consulta->ParamByName("T")->AsString=F_CONS_CLI->NOM->Text;
BD->consulta->Open();
```

En primer lloc, marquem la taula consulta activa com a fals i la tanquem, això es fa perquè en el cas de que hi hagués una consulta bolcada sobre la taula, la para. Tot seguit netegem el codi SQL i seguidament hi afegim la consulta que volem realitzar. En aquest exemple es pot observar que utilitzem un paràmetre per assignar-li al nom, per aquesta raó necessitem la següent sentència, ja que li diem d'on ha de treure la informació. Finalment llancem la consulta amb l'*Open*.

4.4.2.- Disseny en l'entorn web:

Per tal de portar a terme el disseny i la implementació de l'entorn web, hem utilitzat el Dreamweaver MX. Fins llavors, ens era un programa totalment desconegut per a nosaltres.

Anem a veure tot seguit un esquema del que és el disseny arquitectònic de l'entorn web, on tenim l'aplicació i el client:



Quan un usuari accedeix a la pàgina web, s'executa la pàgina amb PHP que es va generar. En aquell moment la pàgina s'omple amb tots els continguts dinàmics interns.

Per tal de realitzar la pàgina web, s'han utilitzat formularis en cada pàgina, amb el qual omplim el que volem consultar, premem el botó i se'ns mostren els resultats obtinguts de la base de dades, hi ha vegades que se'ns mostren a la mateixa pàgina, altres, se'ns obre una nova pàgina. Tot seguit anem a veure el procediment seguit per tal de fer una consulta d'un client, cal dir que s'ha utilitzat una taula dinàmica per tal de fer aquesta consulta, a continuació veurem el codi utilitzat i el resultat obtingut per web:

```

<?php session_start(); ?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Documento sin título</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<style type="text/css">
<!--
.Estilo1 {font-family: Arial, Helvetica, sans-serif}
-->
</style>
</head>
<body>
<span class="Estilo1">
</span>
<form action="" method="post" name="form1" class="Estilo1">
<h1>CONSULTA DE CLIENTS:</h1>
<p>NOM:
<input name="nom" type="text" id="nom" size="40">
<input name="Buscar" type="submit" id="Buscar" value="Buscar">
</p>
</form>
<?php
if (isset($_POST["Buscar"])=="Buscar"){
?>
<table align="center" width="59%" border="0" cellspacing="0" cellpadding="4">
<tr valign="top" bgcolor="#cccccc">
<td align="center" class="Estilo1"><strong>Codi</strong></td>
<td align="center" class="Estilo1"><strong>Nom</strong></td>
<td align="left" class="Estilo1"><strong>Cognoms</strong></td>
</tr>
<?php
include("conexio.php");
$nom_cli=$_POST["nom"];
$sdades_client="select Codi, Nom, Cognoms from persones where Nom like
'".$nom_cli."%'";
$rsdades_client=mysql_db_query($db,$sdades_client)or die mysql_error();
if ($row=mysql_fetch_array($rsdades_client)){
do{
?>
<form method="post" action="resclient.php">
<tr valign="top">
<td align="center" colspan="5"><hr noshade></td>
</tr>
<tr valign="top" bgcolor="#DCDCDC">
<td width="11%" align="center" bgcolor="#DCDCDC">
<?php echo $row["Codi"] ?></td>
<td width="19%" align="center" class="Estilo1">
<?php echo $row["Nom"] ?><br></td>
<td width="53%" align="left" class="Estilo1">
<?php echo $row["Cognoms"] ?></td>
<td width="17%" align="center">

```



```
<input type="hidden" name="codi"
      value="<?php echo $row["Codi"]?>">
<input type="submit" name="consul" value="Consultar">
</form>

<?php
    }while ($row = mysql_fetch_array($rsdades_client));
?></tr>
</table>
<?php
    }else{
        echo "No s'ha trobat ningú amb aquest nom a la base de dades!!";
    }
}
?>
</body>
</html>
```

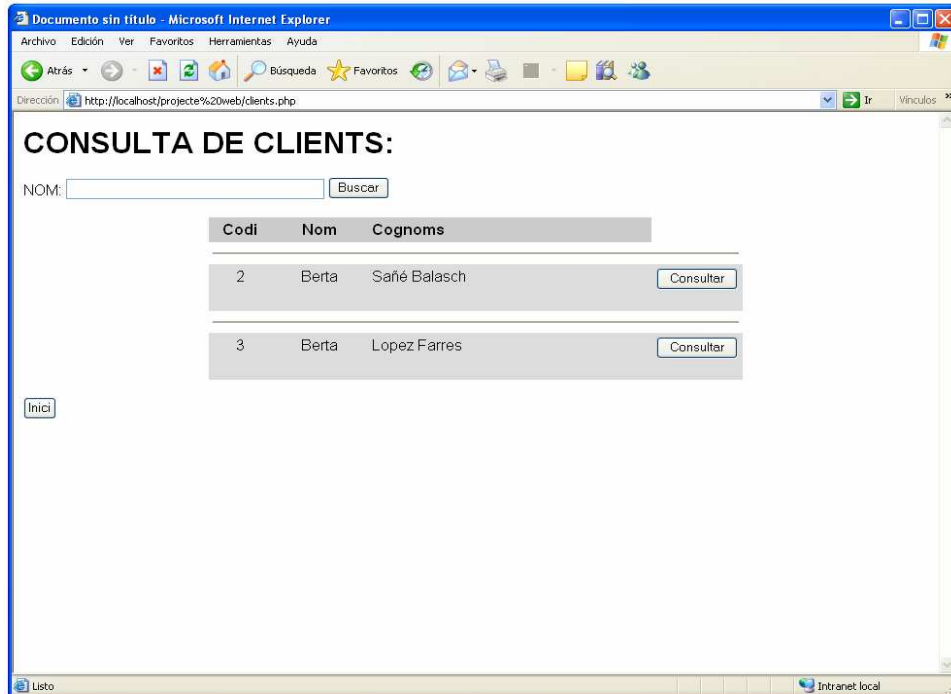


Fig. 1: Consulta de clients de nom 'berta'.

En entrar a consulta de clients, es mostra només la part superior de la figura 1, una vegada escrit el nom i premut buscar es construeix la taula amb les dades buscades. Finalment es prem el botó consultar del registre que es vol i surt la figura 2:

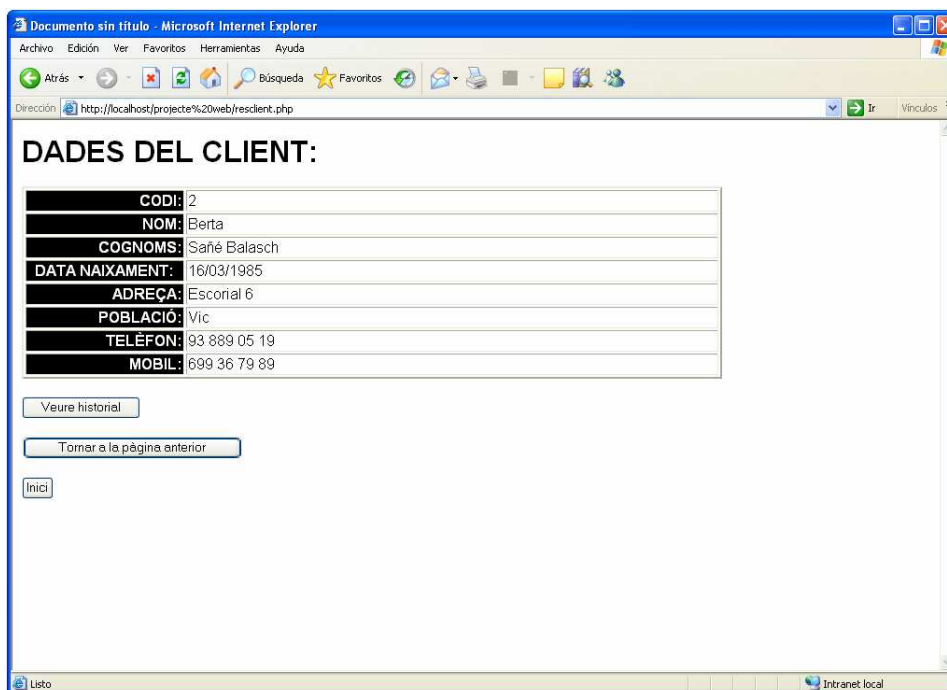


Fig. 2: Dades del client

Una vegada aquí podem consultar l'historial (resultat figura 3) del client o tornar a l'inici (el menú). Cal dir que en el cas de que el pacient encara no s'hagi fet cap sessió, l'historial apareixerà buit.

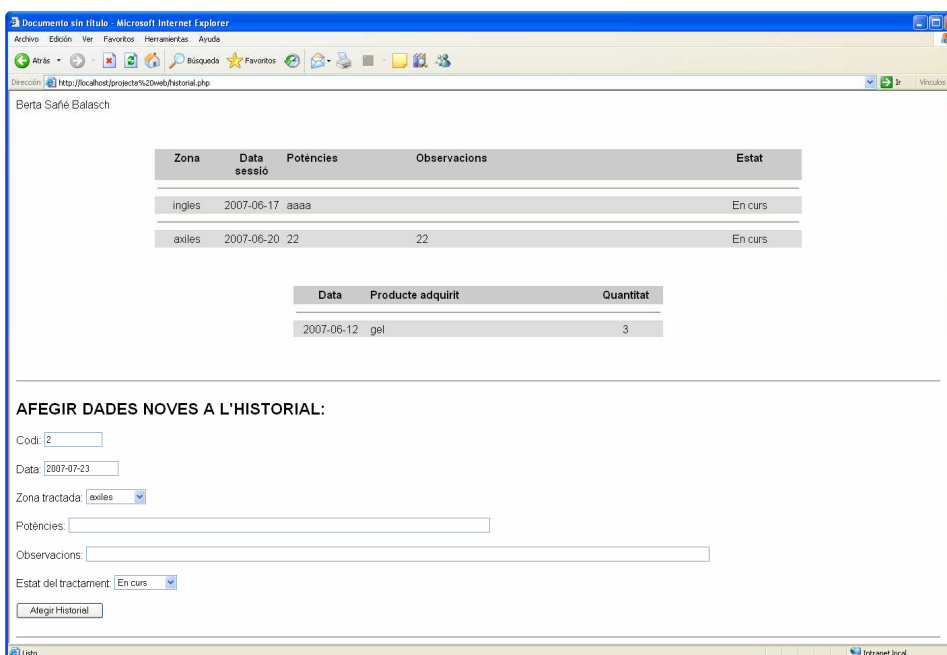


Fig. 3: Resultat de veure historial.

En aquesta pàgina podem veure tot l'historial del client en la part superior, i en la inferior podem entrar les dades d'una nova sessió o un producte comprat.

Finalment pel que fa la connexió a la base de dades des de l'aplicació web, utilitzem un fitxer anomenat conxio.php, que es crida cada vegada que es necessita accedir a la base de dades, aquest fitxes és el següent:

```
<?php
$dbhost="localhost";      // host del MySQL (generalment localhost)
$dbusuari="root";        // nom d'usuari per accedir a la base de dades
$db="projecte";          // Seleccionem la base amb la qual treballar
$conn = mysql_connect($dbhost, $dbusuari) or die ("No conecat a la base de
dades");
mysql_select_db($db, $conn);
?>
```

Una vegada vist el fitxer utilitzat per a fer la connexió, anem a veure la manera de realitzar una consulta a la base de dades:

Consulta utilitzada per tal de generar la factura d'un centre:

```
$sfactura=" select p.Nom, p.Cognoms, h.Zona_tractada, t.Preu, c.Percentatge
from persones p, centres c, historial h, tractaments t
where c.Nom='".$centre."' and
      c.Codi=p.Centre_associat and
      p.Codi=h.Codi and
      h.Fecha between '".$d_ini."' and '".$d_fi."' and
      h.Zona_tractada=t.Nom
order by Cognoms";
$rsfactura=mysql_db_query($db,$sfactura)or die (mysql_error());
```

5.- DISSENY ARQUITECTÒNIC I MANUAL D'INSTAL·LACIÓ:

En un cas normal utilitzaríem un servidor per tal de guardar-hi la base de dades, però en el nostre cas, tenim el treballador autònom, que és l'encarregat d'entrar totes les dades a la base de dades a través de l'aplicació amb C++. Per aquesta raó, necessitem que la base de dades i l'aplicació estiguin ubicades en un mateix lloc. Finalment ja s'ha comentat que aquest necessita fer ús d'un portàtil, ja que té varis llocs de visites. Pel que fa el servidor, serà necessari que sempre estigui engegat a les hores de treball.

La part hardware constarà de l'ordinador que hi hagi en centre d'estètica i un portàtil que portarà el treballador autònom que s'utilitzarà també com a servidor.

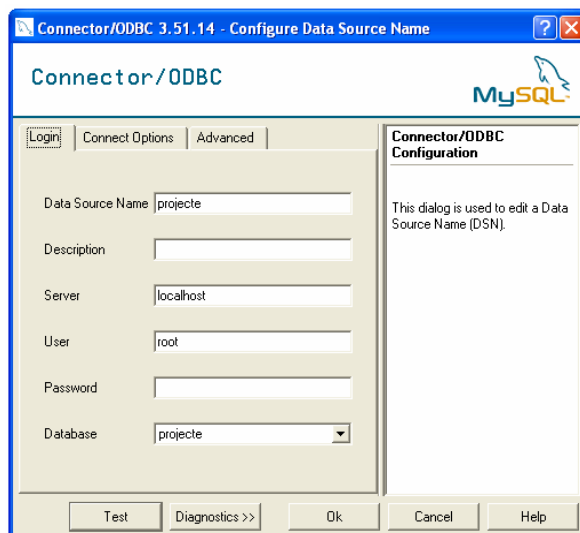
En quan a components software, primer de tot necessitem l'entorn Windows per a treballar. A més, instal·laríem l'aplicació realitzada per a fer tot el control de dades, l'easyPHP, el MySQL Connector ODBC, el DynDNS i el Kaspersky Antivirus Personal que a la mateixa hora ens a serveix com a firewall.

L'easyPHP és un paquet de software en el qual et permet utilitzar el llenguatge PHP, a més de la base de dades. Aquest inclou un servidor Apache, el MySQL, una completa execució del PHP i eines per desenvolupar una pàgina web.

Fent referència al servidor Apache, podem anomenar les següents avantatges: és un software lliure de codi obert, és multi-plataforma, modular, extensible i popular (fàcil d'aconseguir ajuda).

Pel que fa el MySQL, també és un software lliure. És un sistema gestor de base de dades, i en quan a característiques, podem dir que és relacional, amb diferents fils d'execució i multiusuari.

El MySQL Connector ODBC, es necessita per tal de poder crear els drivers i utilitzar l'aplicació. Una vegada instal·lat, s'ha de configurar, per tal d'aconseguir això t'has de dirigir al menú "Inici, Configuración, Panel de Control, Herramientas Administrativas, Origenes de Datos ODBC", i una vegada aquí en la pestanya de "DSN de usuario" agregar un nou origen de dades, on en primer lloc seleccionarem el connector esmentat anteriorment i després omplirem la nova finestra igual que la que es veu a continuació, una vegada fet això, ja et podràs connectar a la base de dades projecte (en aquest cas) des de l'aplicació dissenyada.



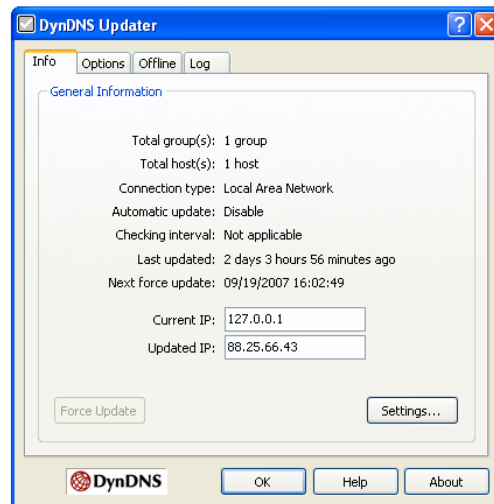
Una altre dels programes que instal·leriem, seria el DynDNS, que l'utilitzarem per tal de convertir una IP dinàmica en una IP fixa. Una direcció de IP és un numero que identifica un dispositiu (normalment un ordinador) dins d'una xarxa que faci servir el protocol IP.

Cada vegada que un usuari es connecta a internet, fa servir una direcció IP. Aquesta pot variar cada vegada, en el cas de que sigui diferent cada vegada que es connecti, vol dir que té IP dinàmica.

Els llocs d'internet que necessiten estar permanentment connectats, generalment tenen una IP fixa o IP estàtica, és a dir, no canvia amb el temps. Els servidors de correu electrònic, DNS, FTP públics i els servidors de pàgines web (el nostre cas) necessàriament han de fer servir una direcció IP fixa, ja que d'aquesta manera la localitzes a la xarxa.

Anem a parlar ara del domini, un domini d'internet és un nom base que agrupa un conjunt d'equips o dispositius, i permet recordar noms d'equips més fàcilment que no pas una IP numèrica.

Per tal de realitzar això, el que has de fer en un primer moment és registrar-te a la pàgina www.dyndns.com. Una vegada registrat vas a configurar el host, aquest é el moment que tens d'escollir el domini que vols en el nostre vas hem escollit "esterlizana.dyndns.org", on el nom de l'equip és "www.esterlizana.dyndns.org". Per acabar ens descarreguem un programa que es diu DynDNS Updater, s'instal·la, i només queda fer la configuració amb el nom d'usuari i la contrasenya que t'has registrat. La única cosa que cal tenir en compte, és que aquest programa arranqui automàticament amb el sistema operatiu (en el nostre cas el Windows). L'aspecte que té el programa en execució és el següent:



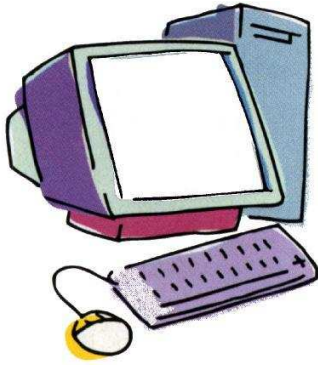
Finalment, parlant del tema de seguretat, necessitem un antivirus i un firewall per tal de protegir l'ordinador, hem optat per instal·lar el Kaspersky Antivirus Personal, que fa les dues funcions alhora.

Un firewall és un element software utilitzat en una xarxa d'ordinadors per controlar les comunicacions. Permet o prohibeix la comunicació segons la política de la xarxa definida per a l'organització responsable. El firewall és el punt de connexió entre la xarxa interna i l'exterior que és internet.

Un firewall correctament configurat afegeix protecció a una instal·lació informàtica, però en cap cas s'ha de considerar suficient. La seguretat informàtica avarca més àmbits i més nivells de treball i protecció.

El Kaspersky Antivirus Personal no és un software gratuït, però és el més complet en quan a software dolent ja que no només s'encarrega dels virus, cucs i troians, sinó que també detecta dialers, espies, keyloggers, entre altres malwares. A més també és el més actualitzat que existeix. Com a inconvenient té la lentitud en ordinadors que no són potents, tot i això Kaspersky disposa d'una base de dades interna que memoritza els arxius escanejats perquè el segon escaneig sigui més ràpid.

També havíem pensat en instal·lar el Ashampoo Firewall, que és un software gratuït, però no és tant potent com el Kaspersky, i a més no té la propietat de ser antivirus a la vegada, per aquesta raó decidim adquirir el Kaspersky Antivirus Personal, per un cost de 59'95 € per ordinador i any.



ORDINADOR DEL CENTRE:

- Kaspersky Antivirus Personal
- Connexió a internet.

Cost: 748 € + 59,95 € (Kaspersky)



ORDINADOR PORTÀTIL + SERVIDOR:

- Aplicació amb C++.
- Sistema gestor de base de dades.
- Base de dades
- Apache
- MySQL Connector ODBC
- DynDNS Updater
- Kaspersky Antivirus Personal
- Connexió a internet.

Cost: 999 € + 59,95 € (Kaspersky)

En el cost que es menciona anteriorment no hi ha comptat el cost del programa. Aquest seria un cost elevat, ja que s'hi han invertit moltes hores de feina, tant en buscar informació i recursos per a realitzar-lo, com en implementar-lo. Ha estat un aprenentatge molt bo de cara a futures aplicacions que puguem fer.

6.- MILLORES I CONCLUSIONS:

En quan a possibles millores del programa realitzat, es podria fer:

1. Tota la gestió del programa a través de la pagina web. Que no fos necessari tenir l'aplicació realitzada amb C++. La raó per a la qual no s'ha realitzat d'aquesta manera, és que d'un principi, resultava ser un únic centre, però una vegada començat el projecte, se'ns va comunicar que tindria més d'un lloc de treball per accedir a la base de dades, ja que tindria més centres. Com que ja estava el projecte en marxa, varem decidir mantenir la part del C++ (que ja estava quasi acabada) i fer la feina restant amb una aplicació web. En quan esforç i cost, creiem que no seria gaire elevat, ja que en l'aplicació web, operacions que realitzes amb el programa amb C++, també s'han implementat, com són els casos de totes les consultes. Només faria falta implementar les altres i les modificacions.
2. Una altre possible millora que es podria realitzar per a l'aplicació, seria la possibilitat de poder avisar d'un dia per l'altre i automàticament, de la visita concertada, via SMS al telèfon mòbil. Aquest és un servei que ofereix telefònica per un preu de 24'95€ mensuals i una quota d'alta de 10€.

Una vegada finalitzat el projecte, hem après a treballar a través de bases de dades sincronitzant-ho amb altres aplicacions com és el cas de la realitzada amb el Builder C++ i també amb la pàgina web. A més som capaços de programar i dissenyar una pàgina web utilitzant el llenguatge de PHP, cosa que fins a llavors ens era totalment desconegut per a nosaltres. D'altra banda, ara ens veiem en cor de muntar un servidor web, de jocs o de FTP en un ordinador qualsevol.

7.- BIBLIOGRAFIA:

1. <http://www.programacion.net/tutorial/php> (Consulta: 20 de maig de 2007)
2. http://www.terra.es/personal/tamarit1/instalacion_servidor/apache/index.html (Consulta: 20 de maig de 2007)
3. http://www.ics-php.com/php_sd/download.htm (Consulta: 20 de maig de 2007)
4. <http://www.webexperto.com/articulos/art/56/instalacion-del-servidor-apache-bajo-windows/> (Consulta: 20 de maig de 2007)
5. http://www.gamarod.com.ar/articulos/conectarse_a_mysql_desde_php.asp (Consulta: 23 de maig de 2007)
6. <http://geneura.ugr.es/~maribel/php/> (Consulta: 5 de juny de 2007)
7. <http://es.wikipedia.org> (Consulta: 15 d'agost de 2007)
8. <http://www.easyphp.org/> (Consulta: 15 d'agost de 2007)
9. <http://www.es.gnome.org/> (Consulta: 16 d'agost de 2007)
10. http://www.ashampoo.com/frontend/products/php/product.php?idstring=0050&session_langid=11 (Consulta: 25 d'agost de 2007)
11. <http://kasperskytienda.com.es/fiche.html?home=1&refProduct=3079&REF=650268> (Consulta: 25 d'agost de 2007)
12. <https://www.dyndns.com> (Consulta: 25 d'agost de 2007)ç
13. <http://www.rafamerida.com/?q=node/75> (Consulta: 25 d'agost de 2007)
14. Apunts de METODOLOGIA DE PROJECTES
15. Apunts de ENGINYERIA DE SOFTWARE DE GESTIÓ – Tema 1 i 3