



HOSTAL.NET

Memòria del Projecte Fi de Carrera  
d'Enginyeria en Informàtica  
realitzat per  
Roger Forcada Bigas  
i dirigit per  
Ana Cortés Fité  
Bellaterra, 14 de Juny de 2007

# Índex

|                                     |                |
|-------------------------------------|----------------|
| <b>1.- Introducció</b>              | <b>Pàg. 1</b>  |
| 1.1.- Motivació del projecte        | Pàg. 2         |
| 1.2.- Organització                  | Pàg. 3         |
| <b>2.- Anàlisis de requeriments</b> | <b>Pàg. 4</b>  |
| 2.1.- Requeriments no funcionals    | Pàg. 5         |
| 2.1.1.- Entorn de funcionament      | Pàg. 5         |
| 2.1.2.- Entorn de treball           | Pàg. 5         |
| 2.1.3.- Gestor de dades             | Pàg. 6         |
| 2.1.4.- Identificació d'usuaris     | Pàg. 8         |
| 2.1.5.- Interacció amb el software  | Pàg. 9         |
| 2.1.6.- Impressió de tiquets        | Pàg. 10        |
| 2.1.7.- Caixers moneders            | Pàg. 11        |
| 2.1.8.- Visors                      | Pàg. 11        |
| 2.2.- Requeriments funcionals       | Pàg. 13        |
| 2.2.1.- Usuaris                     | Pàg. 13        |
| 2.2.2.- Magatzem                    | Pàg. 14        |
| 2.2.3.- Ventas                      | Pàg. 16        |
| 2.2.4.- Tancament de caixa          | Pàg. 17        |
| 2.2.5.- Gestió                      | Pàg. 18        |
| 2.2.6.- Facilitats d'usuari         | Pàg. 18        |
| 2.2.8.- Suport de connexió          | Pàg. 19        |
| <b>3.- Disseny de l'aplicació</b>   | <b>Pàg. 20</b> |
| 3.1.- Resum i esquema general       | Pàg. 21        |
| 3.2.- Estructura de dades           | Pàg. 23        |

|                                              |         |
|----------------------------------------------|---------|
| 3.2.1.- Taula usuaris                        | Pàg. 24 |
| 3.2.2.- Taula caixers                        | Pàg. 25 |
| 3.2.3.- Taula clients                        | Pàg. 25 |
| 3.2.4.- Taula proveïdors                     | Pàg. 26 |
| 3.2.5.- Taula productes                      | Pàg. 26 |
| 3.2.6.- Taula encàrrecs                      | Pàg. 28 |
| 3.2.7.- Taula factures                       | Pàg. 29 |
| 3.2.8.- Taula de factures rebudes            | Pàg. 29 |
| 3.2.9.- Taula extra de dades                 | Pàg. 30 |
| 3.3.- Esquema final de l'estructura de dades | Pàg. 31 |
| 3.4.- Connexió amb el gestor de dades        | Pàg. 32 |
| 3.4.1.- Creació i manteniment de la connexió | Pàg. 32 |
| 3.4.2.- Funcions d'entrada i sortida         | Pàg. 34 |
| 4.- Interfície gràfica d'usuari              | Pàg. 38 |
| 4.1.- Formulari de configuració              | Pàg. 39 |
| 4.2.- Mòdul de suport                        | Pàg. 41 |
| 4.2.1.- Variables globals                    | Pàg. 41 |
| 4.2.2.- Detecció de separador decimal        | Pàg. 41 |
| 4.2.3.- Format fixa dels nombres decimals    | Pàg. 42 |
| 4.2.4.- Carrega de la configuració           | Pàg. 42 |
| 4.2.5.- Sortida per Visor VFD Serie          | Pàg. 43 |
| 4.3.- Formulari principal (Part 1)           | Pàg. 44 |
| 4.4.- Formulari d'ingrés manual              | Pàg. 46 |
| 4.5.- Formulari de control d'usuaris         | Pàg. 48 |
| 4.5.1.- Gravació de targetes                 | Pàg. 49 |
| 4.6.- Formulari Principal (Part 2)           | Pàg. 50 |
| 4.6.1.- Lectura de targetes                  | Pàg. 51 |
| 4.7.- Formulari de control d'estocs          | Pàg. 52 |

|                                                |         |
|------------------------------------------------|---------|
| 4.7.1.- Creació de carpetes                    | Pàg. 54 |
| 4.7.2.- Creació de productes crus              | Pàg. 55 |
| 4.7.3.- Creació de productes de venda          | Pàg. 57 |
| 4.7.4.- Creació de menús                       | Pàg. 58 |
| 4.7.5.- Explorador de magatzem                 | Pàg. 59 |
| 4.7.6.- Formulari detall d'estadístiques       | Pàg. 61 |
| 4.7.7.- Vista detall                           | Pàg. 61 |
| 4.7.8.- Entrada de productes crus              | Pàg. 62 |
| 4.8.- Formulari TPV                            | Pàg. 64 |
| 4.8.1.- Format de factura / encàrrec           | Pàg. 65 |
| 4.8.2.- Gestió d'encàrrecs                     | Pàg. 66 |
| 4.8.3.- Selecció d'un producte de venda o menú | Pàg. 67 |
| 4.8.4.- Canvi i ingrés a caixa                 | Pàg. 69 |
| 4.8.5.- Selecció i gestió de clients           | Pàg. 70 |
| 4.8.6.- Cobrament                              | Pàg. 71 |
| 4.8.7.- Impressió de tiquets                   | Pàg. 73 |
| 4.8.8.- Càlcul de rendiment                    | Pàg. 75 |
| 4.9.- Formulari de quadre de caixa             | Pàg. 77 |
| 4.10.- Formulari de gestió                     | Pàg. 79 |
| 5.- Funcionament bàsic                         | Pàg. 81 |
| 6.- Conclusions                                | Pàg. 86 |
| .- Referències                                 | Pàg. 88 |



# 1 - Introducció

En aquest projecte es mira de donar solució a tots els requeriments que podria tenir un negoci d'hostaleria de menjar ràpid. Lògicament es podrà utilitzar per molts altres tipus de negocis d'hostaleria, i amb poques variacions es podria utilitzar per casi qualsevol negoci que requereixi control d'estocs, venda o facturació sobre una base de dades de productes.

Sempre es mira de donar una seguretat bastant alta, i un control sobre tot el que es pogués dur a terme molt exhaustiu, ja que ara només podrien ser números, files i objectes, però més endavant serien diners, material i persones.

Tot ha estat acordat amb un client final, tot i així, no basaré el projecte amb el desenvolupament per aquesta empresa, tot i que no hi ha cap problema per nombrar-la. El fet es que tot aquest software ha estat desenvolupat per mi com a empresa personal, i el client, el restaurant Chic & Chips ubicat a Diagonal Mar (Centre comercial de Barcelona) només l'utilitza (Ús sobre llicència individual). Com ja comentaré més endavant, molts dels requeriments han estat obtinguts a partir de xerrades amb aquest client final, i les proves de funcionament també han estat fetes en aquest mateix local.

Per últim, afegir que aquest software ja és totalment funcional, i porta alguns mesos funcionant diàriament unes 9 hores. Per tant tots els requeriments es consideren més que assolits i acompanyats. Considerant que la feina de pensar com ha de ser cada apartat del software, i què ha de portar, és només una petita part en comparació a tot el codi que porta al darrere per tal d'assegurar un funcionament correcte del software.

## ► **1.1 - Motivació del projecte**

El projecte neix de la necessitat d'una empresa d'hostaleria d'una solució eficaç i rentable per el seu negoci. En principi es va parlar d'una aplicació senzilla, rapida i de molt fàcil ús, per cobrir les necessitats bàsiques de funcionament d'un restaurant de menjar ràpid. Però a mida que el projecte es va anar desenvolupant, va anar creixent i agafant una forma més solida com a solució integral per el negoci.

Tot el projecte ha estat desenvolupat en conjunt amb el director del negoci i el seu encarregat de manteniment del local, tot per aconseguir tenir una idea més unida entre enginyer i client. Per tal de crear una aplicació més definida, varem fer servir una metodologia de treball per entregues, amb successives quedades per parlar de cadascun dels temes, on es varen anar creant esquemes de funcionament, i agafant idees del resultat que realment s'esperava.

Lògicament, una de les motivacions afegides del projecte també seria l'aprenentatge, no només de les eines de desenvolupament, sinó també de la metodologia de treball d'un enginyer en contacte amb un client. Ja que tenir tot escrit i definit en un full, amb esquemes i explicacions, de com ha de ser una aplicació (Una pràctica, per més gran que sigui) és molt senzill en comparació a aconseguir plasmar la idea que el client desitja amb xerrades i demostracions. Tots sabem que moltes vegades ni el client sap el que necessita, i moltes vegades no sap exactament ni el que demana, el què ens dona una feina suplementària de reflexió i optimització del projecte. Tot sempre per tal de reduir-li despeses al client, i fer un treball ben fet.

## ► **1.2 - Organització**

Tot el projecte serà explicat seguint cada pas que vaig seguir per desenvolupar el software. Primer començant per escollir tot el que necessito per treballar, tot el que necessita el client per treballar. Com necessita el client que es faci el software, i pas a pas, començant per les bases, i acabant amb els detalls, explicant per sobre com s'ha fet cada apartat.

Per tant hi haurà dos apartats principals, el de requeriments (Què es necessita) i el de disseny (Com es desenvolupa). I cadascuna de les seves parts explicada per ordre de prioritats.

Per tal d'oferir una vista més intuïtiva de com he anat fent les coses, inclouré esquemes per explicar el funcionament d'algun procediment concret, i de resums generals. I també inclouré vistes de cada formulari de d'interfície.

En alguns casos les explicacions es podran tornar més tècniques, però a mida que avanci en el projecte, aniré donant més coses per suposat, i la explicació serà més superficial i de funcionament.

## **2 - Anàlisis de requeriments**

Tot l'anàlisi de requeriments, com ja he esmentat abans, ha estat desenvolupat a mida que el projecte avançava, amb trobades, xerrades i demostracions, tot i que també hi ha molts requeriments, els quals l'usuari final no en té cap idea, i tampoc ha de tenir-ne.

Per tal d'estructurar bé i diferenciar cada tipus de requeriment, els diferenciaré en requeriments no funcionals (Tècnics o de material), i requeriments funcionals.

En els requeriments no funcionals parlaré de les infraestructures que caldria utilitzar, el material, entorns de treball, tipus de gestió de dades i tot lo relacionat. I en els requeriments funcionals esquematitzaré el model funcional de l'aplicació, pas a pas, segons cadascuna de les necessitats del client.

Cal tenir en compte també que aquest software continua en constant desenvolupament, el desenvolupament a petició del client és completament en castellà, i podria ser que en alguna captura de pantalla, o en la versió entregada, apareguin algunes opcions addicionals no descrites en aquest projecte, ja que es troben en fase de proves, i que les qualificaríem de complements opcionals, però no com a requeriments.

## ► **2.1 - Requeriments no funcionals**

Primer de tot, tenim dos requeriments bàsics a resoldre des del punt de vista del client: l'entorn de funcionament de l'aplicació, el que ens limitarà l'entorn de treball, i després d'igual d'importància tenim la gestió de dades.

### **- 2.1.1 - Entorn de Funcionament**

El client en tot moment parlava d'una cosa senzilla i de ràpida instal·lació, amb equips que es trobin al mercat fàcilment, sense necessitat de tècnics addicionals per la instal·lació, i que ofereixin garanties, per tal de no tenir problemes en cas de que la maquinaria fallés. Directament degut als requeriments introduïts pel client, tindrem que eliminar totes les plataformes gratuïtes ( com poden ser Linux i similars) les quals requereixen una instal·lació i manteniment més complicats, que molt probablement hauria de fer jo mateix o un tècnic ( més despesa), i casi també faria falta un tècnic per el manteniment. Això ràpidament ens fa pensar en buscar la solució informàtica més venuda a tot el mon, d'estabilitat i funcionalitat discutible, però d'un ús ràpid, eficaç i molt adaptable, el conegut Microsoft Windows, en tal cas per les versions que tenen al mercat actualment, faríem servir un Windows XP Home ( el més senzill) per els equips que portaran el software.

### **- 2.1.2 - Entorn de treball**

Un cop escollit l'entorn de funcionament, abans d'escollir el sistema gestor de dades, crec que és millor escollir l'entorn de treball, ja que en molts casos és molt important el connector entre el llenguatge o aplicació de programació, i el gestor de dades, en el cas que sigui extern, i en el cas que sigui intern doncs també és molt important saber amb anterioritat les funcions de les que disposaràs per gestionar-ho

tot i les limitacions que puguis tenir. Per tant, ja sabem que l'aplicació funcionarà amb Windows, tot i així, això no ens limita a treballar en Windows, ja que per exemple una aplicació Java la podríem desenvolupar en altres plataformes i instal·lar-la després on desitgem, però no ens interessa el Java, ja que volem alguna cosa més senzilla d'instal·lar i de posar en marxa, i alhora també molt més visual i pràctica, que no ens fes perdre el temps en desenvolupar massa codi, ni en instal·lar llibreries manualment, i directament es puguin fer proves tant en el meu entorn, com en el de destí. Després d'analitzar diferents entorns i discutir-los amb el client, el que més ens va convèncer als dos va ser el Visual Basic.NET, el codi és mínim, és molt visual, té fàcil gestió d'errors, les llibreries són totes d'instal·lació automàtica (Tot el que requerim pot estar en llibreries .DLL que només s'hauran de copiar al directori destí), i l'entorn .NET Framework 2.0 que serà el de treball del Basic.NET, es descarrega automàticament per les actualitzacions Windows Update, per tant, més facilitats impossible. La única cosa que ens preocupava d'escollir un Visual Basic.NET era el fet de ser un entorn comercial, i no gratuït com el Java o d'altres, però degut a la projecció de futur de la meua empresa com a desenvolupador de software, una llicència de desenvolupament de Visual Studio Professional no és tanta despesa si fas un càlcul ràpid del temps que t'estalvies en altres coses, ja que en tot moment sempre ens basem en tenir una bona relació entre el temps invertit en crear la solució, i el resultat que dona, per tant només ens interessa el rendiment.

### **- 2.1.3 - Gestor de dades**

En tercer lloc tenim la elecció del sistema gestor de dades. Parlant amb el client, es va comentar ràpidament que la informació que necessitava guardar era bastant extensa, d'actualització molt freqüent, i amb la necessitat imperativa de crear

un sistema centralitzat de dades, ja que tindrem varis caixers treballant alhora amb un mateix stock, entre d'altres coses. Degut a això, el sistema gestor de dades ja passa a ser obligatòriament, elecció de quin sistema de base de dades utilitzem, eliminant tot sistema de gestió de dades sobre arxiu directe, en binari o ASCII. Per tant, i tornant al que comentàvem en primer lloc, hem d'elegir una base de dades senzilla i versàtil, de fàcil instal·lació sobre una plataforma Windows ( No obligatòriament, però per donar facilitats al client estaria molt be) i que alhora sigui de gran capacitat, i consultes, instruccions i transaccions segures i ràpides (com a transacció entenem : conjunt d'instruccions executades en un estat no variable de la base de dades ).. Les primeres propostes podrien ser el Microsoft SQL, PostgreSQL o MySQL. Totes compleixen el format SQL estàndard per facilitar les consultes, i totes tenen una gran capacitat. Probablement la de més capacitat seria MicrosoftSQL, la qual ofereix moltes garanties, i la instal·lació és realment senzilla ( tot seria Microsoft), però tot i així no la utilitzarem degut al seu gran cost de llicència que seria despesa del client, al qual no li interessa, al igual que qualsevol altre solució comercial. Per tant ens quedem amb les solucions no comercials més bones, PostgreSQL i MySQL, les dos actualment ja funcionen en entorn Windows, però en el moment de començar el projecte, PostgreSQL no disposava d'instal·lador ni executables per Windows, i de connectors en fase de proves per .NET 2.0, per tant l'elecció va ser fàcil, MySQL Community Server, que disposa de varies versions de connectors .NET 2.0 en model productiu des de fa molt temps. Tot i així, PostgreSQL i MySQL són bases de dades molt semblants i amb característiques molt iguals, el fet que fa que l'elecció normalment sigui més per facilitat d'ús del programador, entorn ( com el nostre cas), o temes més superficials que no una altra cosa.

#### **- 2.1.4 - Identificació d'usuaris**

Un cop que ja sabem com treballarem, existeixen altres requeriments tècnics, primer de tot el client desitjava un sistema d'identificació de personal còmode i eficaç, sempre amb la opció de la típica identificació d'usuari i contrasenya, però només per casos concrets. Per tant necessitem escollir un suport físic on guardar la informació de cada usuari, que sigui fàcil de fer llegir pel software, i que alhora no sigui massa car. Les opcions que teníem per escollir eren moltes, però totes amb algun problema o limitació, o amb un preu massa alt. Primer varem comentar de fer servir targetes de banda magnètica, com pot ser qualsevol targeta bancària. Però resulta que el gravador de targetes era molt car, i tot i ser les targetes molt barates, com a nosaltres no ens interessa tenir-ne moltes (ja que com a molt podrien ser 10 treballadors), no era una bona elecció. L'altre elecció que ja va ser la definitiva, varen ser les targetes Chip SmartCard, concretament amb el model de chip 4442, molt semblants a les targetes moneder dels bancs. La capacitat que tenen es de 256 bytes, es poden bloquejar d'escriptura o de lectura per diferents codis PIN, són bastant barates, i tots els lectors poden gravar i són barats també.



*Figura 1*



*Figura 2*

*Figura 1: Teclat amb lector/gravador de targetes chip genèric model Chipdrive Keyboard Pro, de la marca alemanya Chipdrive.*

*Figura 2: Banda inferior d'una targeta chip 4442, visualitzant els contactes corresponents a la connexió.*

### **- 2.1.5 - Interacció amb el software**

Aquí tenim que escollir i parlar de com es faria anar el software. Ràpidament quan parlem d'un entorn Windows tothom pensa en un ratolí i un teclat. Però ràpidament es veu que en un negoci d'hostaleria i de cara al client del restaurant, queda molt millor i molt més professional, usar pantalles tàctils [1], i gràcies a la identificació amb suport físic dels usuaris, poder eliminar per complet el teclat i el ratolí. L'elecció aquí ja era més cosa del client que meva, ja que tots els monitors tàctils funcionen amb controladors de Windows que emulen un ratolí, i ràpidament vaig aconsellar-li que comprés el que desitgés però de marca 3M actualment pioners en el sector de solucions d'aquest estil per entorns industrials i d'hostaleria.



*Figura 3*



*Figura 4*

*Figura 3: Part davantera del Monitor Tàctil 3M model M150, on podem apreciar la forma de treballar del monitor i la seva aparença.*

*Figura 4: Part trasera del Monitor Tàctil 3M model M150, on podem observar les connexions, com s'organitzen, i com és el suport central per ajustar la posició. Començant per l'esquerra tenim: Connexió USB/Mouse tàctil, micròfon, altaveus, corrent, VGA.*

### **- 2.1.6 - Impressió de tiquets**

Com a qualsevol negoci, en qualsevol venda ha d'existir una factura per el client en algun format legal, per tant és important també parlar de quina impressora serà la que crearà aquests tiquets, ja que això pot limitar-ne el format, el qual el tinc que dissenyar tot jo, hi ha de ser genèric. Buscant per el mercat ràpidament trobem que hi ha moltes solucions ja preparades per negocis d'hostaleria, sobretot les empreses Epson i Samsung tenen moltes impressores de tiquets tèrmiques ( d'un alt rendiment i molta velocitat), que ja venen preparades inclús per endollar caixers moneders. El seu preu es una mica alt, però tenen una durabilitat molt alta també, el manteniment és mínim, i el paper es molt barat. Comparant una mica els models, per garanties, controladors y anys en el sector, ens quedem amb les Epson TM [2] , una sèrie molt vista en molts locals.



*Figura 5*



*Figura 6*

*Figura 5: Part davantera de la impressora tèrmica Epson TM-T88IV dissenyada per negocis amb molt volum d'impressió.*

*Figura 6: Part trasera de la impressora tèrmica Epson TM-T88IV, podem observar la connexió USB d'impressió, la RJ11 per connectar un caixer moneder, i la de corrent.*

### **- 2.1.7 - Caixers moneders**

Lògicament per cada venda, gestionarem diners, o tiquets de targetes bancàries, i per tant haurem de tenir un caixer on guardar-les. El caixer podia ser de qualsevol tipus, color o marca, però amb la necessitat de que el software el reconegués i pogués tenir-lo tancat, o obrir-lo segons fos necessari. Per tant, tenim la facilitat de que la impressora que hem escollit ja disposa de connector per caixer moneder, per tal d'obrir-lo amb l'enviament d'un sol codi ASCII, per tan el caixer pot ser un qualsevol amb connexió RJ11. En el nostre cas hem agafat un genèric de color negre compatible amb l'impressora, amb ranures per guardar tot tipus de monedes, i separadors per tiquets targeta i tiquets restaurant. Concretament 9 per monedes, 5 per bitllets, amb unes dimensions de 425x425x90mm, amb ranures interiors inferiors per tiquets, tot desmuntable i amb clau de seguretat.



*Figura 7*



*Figura 8*

*Figura 1 i 2: Caixa moneder TPV genèric POS-.430 amb connexió RJ11 per obertura automàtica.*

### **- 2.1.8 - Visor**

També hem de comptar amb que per agilitzar les ventes, el client ha de poder veure el preu, i el canvi en un visor gran i clar, abans inclús de que el venedor li tingui que comentar. Per tant tenim que escollir un visor. Amb el visor estem casi igual que amb el calaix, hi ha molts models, tots més o menys iguals. Normalment treballen directe a Port Serie RS232 amb connexió amb controlador COM, i

s'utilitzen i manipulen amb una connexió estàndard binària 9600 pel port. Per tant igual que en el cas del calaix, hem agafat un genèric que s'adapti a les nostres necessitats, i que tingui un preu reduït.



*Figura 9*

- Visor TPV VFD genèric de 2 línies per 20 caràcters per línia -

## ► 2.2 - Requeriments funcionals

Com a requeriment funcional entenem: les funcions o tasques que es requereix que el programa pugui dur a terme. Per tant és una explicació del que el software pot fer, vist des del punt de vista del client. Molts dels requeriments funcionals aniran vinculats amb els requeriments tècnics, lògicament ja que per dur a terme algunes funcions del software, hem utilitzat el hardware.

### - 2.2.1 - Usuaris

Hem comentat dels usuaris, que necessitàvem un mètode d'identificació per posterior us del software que sigues robust i amb suport físic, i ja hem descrit que el suport físic seran les targetes chip. Per tant necessitem que el nostre programa pugui gestionar una base de dades de clients, vincular-la amb algun enllaç a les targetes, i que apart d'això pugui diferenciar el nivell dels usuaris per poder retallar possibilitats segons càrrec. El qual inclou un sistema de protecció per tal de prevenir que el software pugui ser usat, sinó has ingressat com a usuari corresponent a les tasques que es vulguin dur a terme. Al client també li interessava molt poder tenir un historial d'aplicació per tal de saber exactament que ha fet cada usuari en cada moment, i evitar problemes a l'empresa. Apart també de garantir d'alguna forma que les targetes no podran ser copiades. A la *figura 10* tenim un exemple funcional esquemàtic:

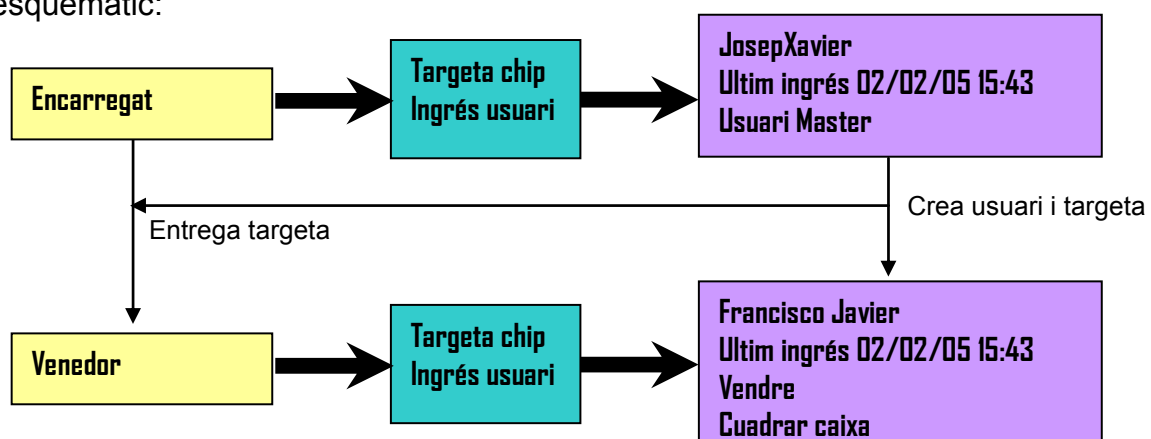


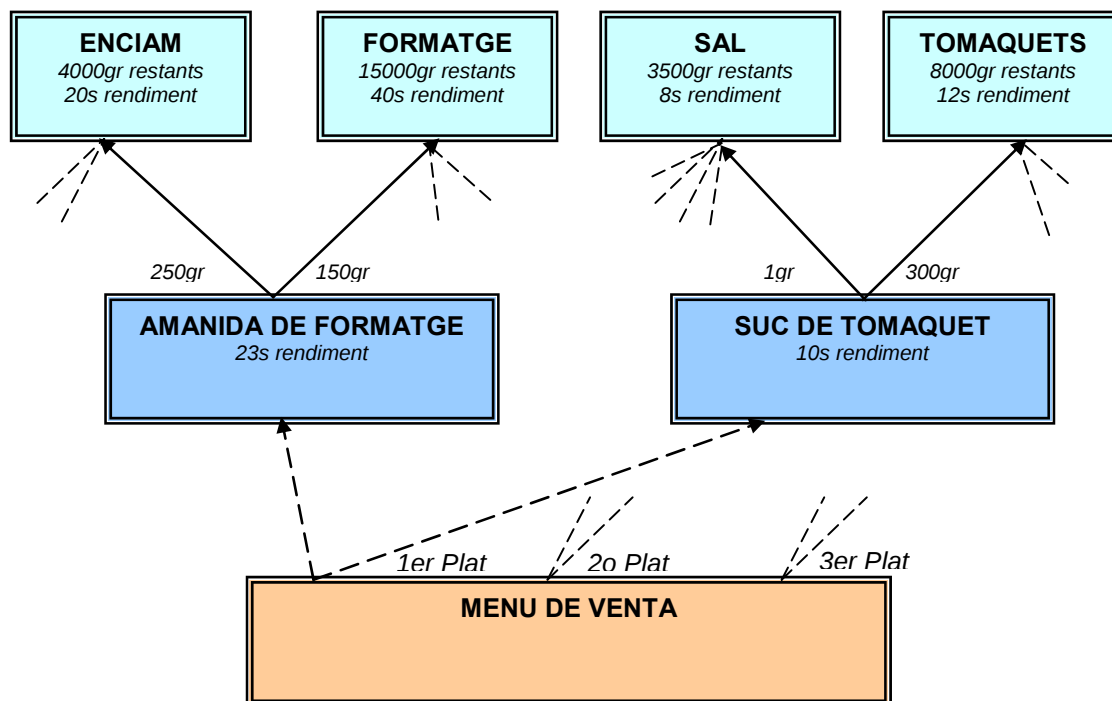
Figura 10

### **- 2.2.2 - Magatzem**

En el magatzem tenim molts requeriments funcionals, molts imposats per el client, i molts extrets per lògica. Hem de tenir un control d'estocs, d'entrada i sortida, per poder gestionar cada un dels materials o productes que el client desitgi poder vendre, i alhora necessita una estimació d'esgotament de cadascun dels productes. Fet que ens porta directament a necessitar per obligació a numerar amb alguna unitat cadascun dels productes, i a realitzar un càlcul actualitzat a cada venda del rendiment de cada producte. Però tot això es complica molt més quan sabem que cada producte de venda, pot ser un got de cocaola amb gel, o pot ser un plat combinat amb patates fregides, sal, un ou ferrat, ketchup i maonesa. Ja que al client li interessa controlar el estoc tan de la cocaola, com del gel, les patates fregides, la sal, els ous, el ketchup i la maonesa i a molt estirar de l'oli de la fregidora. I el client també vol que es puguin fer pacs, ofertes, menús, i que els costs de cadascun es puguin veure al moment de gestionar el magatzem, per tal de poder fixar un PVP correcte. Per tant com a requeriment ja tenim directament, una diferenciació entre productes en cru, productes de venda o pacs ( aquí englobarem les ofertes ja que és el mateix amb preu reduït) i productes menú, els quals han de poder deixar escollir entre productes de venda per cada plat, i el número de plats ha de poder ser variable i amb suplement de preu variable si es correspon, i tot ha de ser còmode per prendre nota al moment de la venda. I tot el que hem comentat fins ara ha d'estar vinculat, per tal de poder tenir un càlcul bo de rendiments i esgotaments. Els productes en cru seran productes de unitat fixada, on tindrem el càlcul de rendiment bàsic de com es gasta cadascun dels materials. Càlculs els quals s'expandiran inversament des dels productes que si es poden vendre, per tenir un càlcul aproximat del rendiment, o més ben dit, de la freqüència de venda, ja que amb el càlcul de freqüència de venda, i

la quantitat restant, podrem esbrinar ràpidament el temps que falta per l'esgotament. Per freqüència de venta entendrem el temps que passa entre venta i venta de una sola unitat de cadascun dels productes, en el cas dels productes de venta, seria per exemple cada quan s'ha venut un Plat combinat N°1, però en els productes en cru seria per exemple cada quan s'ha venut 1gr de carn, cada quan s'ha venut 1gr. d sal, 1 ou, 1 gr. de ketchup, etc.... O utilitzant les unitats que el client desitgi mentre organitza el magatzem, ja que això volem que sigui variable.

Un exemple bàsic de l'estructura que desitgem, una mica retallada, es pot observar a la *figura 11*.



*Figura 11*

Per tant, amb una idea de com s'ha d'estructurar el magatzem, ja sabem que necessitarem poder crear productes crus, productes de venta, i productes menú, i necessitem que es puguin gestionar còmodament, i organitzar per categories. Apart hem de tenir un lloc per veure les dades i càlculs de rendiments, i d'altres a simple vista, per avisar, amb un menú apart per veure-ho més detallat. També necessitem

controlar la entrada de producte cru, per factures, proveïdors i quantitats, per tal de gestionar tema econòmic. Per tant necessitem un historial de facturació d'entrada, i una base de dades de proveïdors.

### **- 2.2.3 - Ventas**

En aquest apartat fem referència als requeriments que s'han de complir, i que és tot el que envolta i es necessita per fer una venta. D'entrada sabem que necessitem un control d'estocs que hauria de gestionar l'entrada, els càlculs de venta i preveure l'esgotament, ara hem de parlar de tot el que envolta el tràmit de venta. Per tal de poder vendre necessitem un apartat de l'aplicació que sigui TPV, que pugui ser tot gestionat amb el monitor tàctil, i un explorador totalment relacionat amb el magatzem. L'explorador per suposat ha de ser tot amb imatges i descripcions, però alhora també hem de poder veure el nom i el preu. Apart de que ha de diferenciar entre un producte de venta normal, o un menú, ja que el menú s'ha de poder seleccionar què voldrà menjar en cada plat (1er, segon, postres,etc...) dels que l'encarregat de magatzem hagi decidit que poden compondre aquell menú.

Un cop tenim tot l'explorador i tenim la forma de escollir que voldrem vendre, necessitem tenir un control dels productes que estan apunt de ser venuts, apart de que també necessitem poder tenir factures, ja venudes, però pendents de cobrar ( encàrrecs i enviaments). També necessitarem poder veure un historial del que s'ha venut, poder gestionar la venta actual còmodament ( treure i posar productes o crear i esborrar preventas), així com també gestionar la caixa ( ingressar o treure diners i poder agafar canvi ), seleccionar clients per facturar ( si es dona el cas necessari ), i per últim un sistema de cobro i el corresponent tancament de caixa. I tot ha d'anar enllaçat de tal forma que si es crea alguna preventa el material d'estoc ha de quedar

reservat, i si s'anul·la s'ha de retornar, però només en el cas que es vengui i es cobri s'han de realitzar-se els càlculs de rendiment.

També serà molt important d'estacar el funcionament incorporat del Visor VFD per port serie, que hauria de mostrar en cada moment el preu de la última unitat venuda, i al final, quan ja es cobra, el total, els diners entregats i el canvi. I un cop tot això funcioni, i ja s'hagi venut un pedido, necessitarem que el calaix s'obri per poder agafar el canvi, i s'imprimeixin els tiquets corresponents a la factura, amb tot detallat, el logotip de l'empresa, i el número de copies desitjat.

#### **- 2.2.4 - Sistema de cobrament**

Aquest sistema ha de ser principalment pràctic i còmode per cobrar a la gent, ha de mostrar-se com una calculadora, però amb la limitació que sabem que sempre cobrarem una xifra en euros, per tant és un número de 2 decimals fixats. També hem de poder seleccionar si volen pagar amb targeta, efectiu o tiquets restaurant. O també poder pagar una factura barrejant formes de pagament (2 euros amb efectiu, 5 amb targeta i 2 tiquets restaurant), per facilitar el cobrament a grups de gent, i fer tots els càlculs corresponents al canvi. Guardant tota la informació del cobrament a la base de dades.

#### **- 2.2.5 - Tancament de caixa**

Ha de ser una pantalla on es mostri detalladament tota la informació de la caixa, com l'últim tancament, usuari d'últim tancament, però només en el cas de ser un encarregat s'ha de poder visualitzar l'historial ( ja que sinó es podria tancar caixa a simple vista i sempre quadraria), s'ha de poder introduir de cadascun dels tipus de cobrament, les quantitats que es troben a caixa, per tal de poder quadrar amb les

dades de la base de dades i fer-ho quadrar tot. Tenint pensat que aquest apartat s'ha de fer als caixers obligatòriament, i per tant ha de ser tot còmode amb el monitor tàctil.

#### **- 2.2.6 - Gestió**

L'encarregat del local ha de tenir molt a mà poder treure un historial general molt detallat de totes les entrades i sortides efectuades en el seu local entre els períodes que ell desitgi. El format està per escollir, però sabem que ha d'anar ordenat per data, hi ha de portar tota la informació que engloba cada venda, i cada compra. Ha de ser àgil, i per na be s'hi hauria de poder explorar, per visualitzar millor que era cada factura, ja que un llistat en algun cas no seria suficient.

#### **- 2.2.7 - Facilitats d'usuari**

Com a facilitats d'usuari podem tenir coses tan simples com interfícies per poder introduir números en caselles al magatzem o al tancament de caixa amb el monitor tàctil, o el fet de que es pugui instal·lar i configurar ell mateix el software.

Com sabem que serà un sistema distribuït, hem de tenir en compte que existiran unes connexions, les quals hauran de ser per obligació segures amb usuari i contrasenya, i en un host apart. Per tant necessitarem un apartat on puguem configurar o retocar tota aquesta informació per tal de que el programa un cop instal·lat es pugui configurar amb molta facilitat, així com també un lloc on configurar coses suplementàries, com el logotip que s'imprimirà als tiquets, o si volem imatges al explorador de magatzem i TPV ( amb imatges implica bastant ample de banda), i més coses que puguin sortir.

#### **- 2.2.8 - Suport de connexió**

Aquest requeriment té a veure amb tot el que envolta la connexió del nostre software amb la base de dades corresponent en el host desitjat. No és un requeriment imposat per l'usuari final, però és un requeriment funcional molt important, no només per l'usuari per tal de que el software tingui una gestió més estable de les connexions amb el host central, sinó també per nosaltres a la hora de desenvolupar l'aplicació, ja que pot facilitar molt moltes de les tasques de desenvolupament, tenir un control molt més precís de les connexions, i poder-ne tenir un control d'errors fiable. Això implicarà crear un mini controlador, dedicat només a la nostre aplicació, on farem totes les peticions que desitgem de la base de dades.

## 3 - Disseny de l'aplicació

L'explicació del disseny de l'aplicació, al contrari que el dels requeriments, serà totalment estructurat en funció de les coses més necessàries en cada moment, de tal forma que començaré per la base del programa i aniré seguint tapant cadascun dels requeriments a mesura que avanço.

Al principi explicaré una mica de codi perquè s'entengui com es dissenya l'aplicació, la forma de treballar, però en general més endavant no explicaré res més, i donaré tot el que ja hagi explicat per suposat. Però en el cas de que utilitzi alguna llibreria externa, que no sigui del .NET Framework 2.0, o no sigui llibreria comú, explicaré què i com la he utilitzat.

### ► **3.1 - Resum i esquema general**

Primer de tot hem de crear un esquema de com volem que sigui la nostre aplicació, per tal de tenir després cadascuna de les parts més estructurada per atacar-les per separat. I tot això ens sortirà directament dels requeriments, principalment dels funcionals, i complementant amb els tècnics.

En el llenguatge Basic.NET desenvoluparé l'aplicació en classes i mòduls, tot contingut dins el projecte Hostal.NET ( El software està desenvolupat en castellà a petició del client). Cada classe es un objecte que pot contenir el que desitgi, la classe més estàndard és el formulari, en el qual es desenvolupen totes les interaccions i successos amb l'usuari ( Entorn visual, interfície i gestió de successos de perifèrics). També utilitzaré alguns mòduls, per acumular funcions genèriques que s'utilitzen molt freqüentment, i alguna classe per desenvolupar controls millorats ( com pot ser el cas de la connexió amb la base de dades).

Per tant, tot el projecte podrà quedar definit segons formularis, classes pròpies, mòduls i agents externs. Agents externs poden ser llibreries d'ús d'algun perifèric, o les bàsiques del llenguatge ( .NET Framework 2.0), ja que només s'utilitzen les que nosaltres desitgem ( sempre hi ha un pack bàsic, i més d'extensos per altres utilitats ), però en principi no ens caldran.

Un cop tenim clar com hem d'enfocar tot el desenvolupament, hem de començar a fer un esborrany de tot el que l'ha de compondre. I ja que el client necessita tenir una idea de com serà l'aplicació, primer ens centrarem en mostrar un esbós del que hauria de incloure. Un esquema funcional d'interfície es pot observar a la *figura 12 a la següent pàgina*.

:

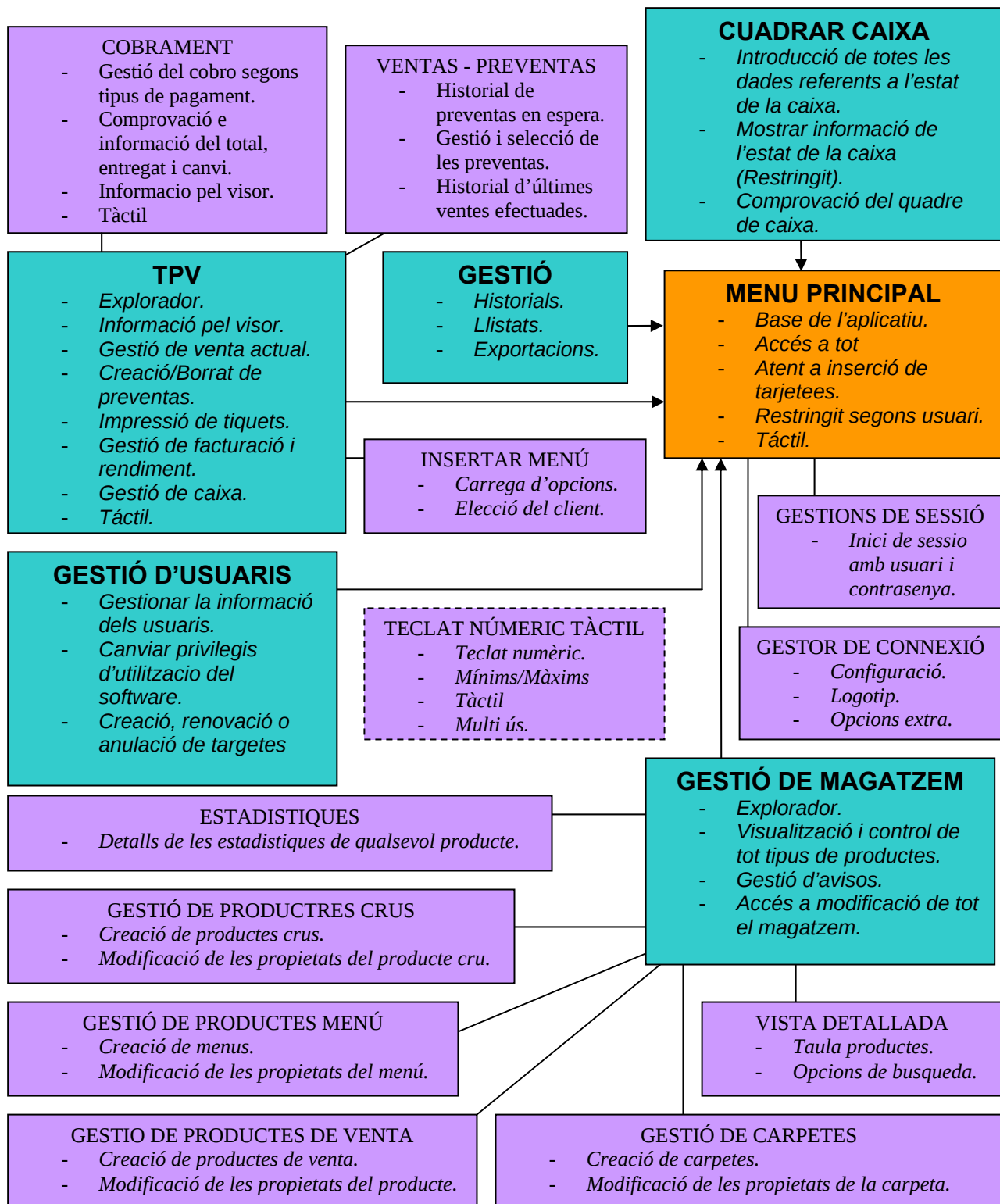


Figura 12

L'esquema de la figura 12 serviria també de cara al client, perquè pugues tenir una idea general del que està demanant, i la funcionalitat que tindrà. Després ja encararem cadascun d'aquests apartats per separat.

## ► **3.2 - Estructura de les dades**

De moment ens cal resoldre una altra cosa molt més important, la connexió amb la base de dades i l'estructura de la mateixa. Com sense connexió no fem res, ni sense estructura tampoc, no importa la que fem primer. Per tant, primer crearem l'estructura, ja que també és un tema a discutir molts cops amb el client o usuari final.

Estem utilitzant un sistema MySQL, però el format de gravació de les bases de dades, la configuració i tot lo demás no ens importa ara. Només mirarem l'estructura de la base de dades.

Com qualsevol altre, dins d'una base de dades, funciona amb taules, cadascuna amb uns camps de molts tipus diferents a escollir, i tot es gestiona per consultes SQL. Al contrari que els formularis, això no surt tan directe dels requeriments, ja que tot és bastant a escollir per mi, segons com vulgui programar jo el, i com vulgui fer-ho servir tot. Però escollir una bona estructura, apart d'aclarir tot molt, pot fer que sigui molt més fàcil el desenvolupament o qualsevol posterior modificació.

Per tant, primer de tot, hem d'escollir quines taules formaran la nostre base de dades, després hem d'escollir quins camps inclourà cadascuna, de quin tipus seran, i per últim les relacions entre taules. Encara que les relacions no són obligatòries (si el programa està ben fet no importen per res), poden ajudar a garantir la integritat de la base de dades, prevenint-nos de problemes desconeguts o funcionaments incorrectes no previstos del software.

Començarem per descriure l'estructura de les taules. En molts casos serà obvi les raons per afegir algun camp, per tant només explicaré els camps que cregui més decisius o conflictius.

### **- 3.2.1 - Taula usuaris**

Per els usuaris primer de tot tenim les dades bàsiques, la informació personal de cada usuari. Contemplarem el nom, el DNI, la direcció i el telèfon. Seran de tipus 'text' ja que són descriptius, i el que continguin a mi no m'interessa per res, només al propietari o als mateixos empleats. També necessitarem el nivell, i òbviament un camp 'id' que faci de clau primària de la taula, ja que no ens interessa utilitzar el DNI o el nom com a clau primària, ja que volem una total llibertat a la hora d'introduir les dades. No ens interessa cap mena d'integritat de dades aquí, ja que dificultaria la inserció d'usuaris, i nosaltres volem facilitat i velocitat.

Després de tots aquests camps bàsics, tenim alguns ja més conflictius. Degut a que com a requeriment funcional tenim la necessitat d'un suport físic, en format de targetes xip, i requerim que no puguin ser copiades, no en tindrem prou amb els camps que tenim per fer-ho funcional. Per tal de que no puguin ser copiades, ni modificades, ni es pugui fer cap tipus de mal ús de les targetes, prendré dos mesures. Primera, utilitzar números de seguretat, creats aleatòriament cada cop que és crea una targeta, que identificaran l'usuari corresponent. I en segon lloc, cada cop que grabem una targeta, li canviarem el codi PIN per un de nou, i el guardarem a la base de dades. D'aquesta forma, la targeta no podrà ser llegida en cap altre lector de cap forma, i en el cas de que poguessin, no podrien fer cap ús de les dades, ja que són aleatòries, i encara tot i així necessitarien recuperar quin era el PIN de l'antiga targeta, ja que també es comprovarà, i també és aleatori a cada gravació. Per tant necessitarem dos camps més per cada usuari, el camp de seguretat que l'identificarà a la targeta, i el codi PIN de la targeta única que li correspon tenir per ingressar.

### **- 3.2.2 - Taula caixers**

Els requeriments marquen que hem de tenir una visió general de l'estat del caixer amb només mirar les dades de la base de dades. Apart de que ha de permetre comprovar els tancaments de caixa.

Per tant per tenir una idea del estat, necessitem un camp del total de cobraments amb targeta, un altre per el total de cobraments amb tiquets restaurant, un del total de diners en efectiu que conté, i un últim per saber quants d'aquests diners en efectiu són deixats de l'últim tancament com a canvi.

Apart de la informació del estat ens interessa també una mica d'història, per tant voldrem un camp amb el codi d'usuari de l'últim tancament de caixa, i un altre per la data dia/mes/any HH:MM:SS.

I per últim només queda decidir quina serà la clau primària. Es va discutir posar un camp 'id' com en els usuaris, però això no ens interessava massa. Ja que els caixers han de saber qui són dins de la base de dades, i si no s'han fet funcionar mai, hauran de fer-hi el seu furat. Per tant varem decidir utilitzar com a clau primària el nom del host en xarxa del ordinador que farà de caixer. Així no farà falta configurar res, inclús ni si reinstal·lem l'ordinador, o el canviem, només posant el mateix nom ja agafarà totes les dades.

### **- 3.2.3 - Taula clients**

Aquesta taula només serà utilitzada en cas d'encàrrecs o ventes a empresa. Per una banda necessitaríem totes les dades personals per poder fer l'enviament, o per garantir que no és estafa ( Trucades telefòniques etc...), i per altre banda necessitaríem totes les dades per poder facturar. Per tant en els dos casos

necessitarem les mateixes dades, un nom, un dni/nif, una direcció i un camp detalls on deixarem escriure tot el que es desitgi ( Telèfon, informació addicional etc...). Apart de una clau primària 'id' totalment independent, per si es dona el cas de que volguéssim donar d'alta un client del qual desconexim les dades, però volem saber-ne la quantitat de compres, per efectuar regals, ofertes o el que sigui...

#### **- 3.2.4 - Taula proveïdors**

Aquesta taula podria ser exactament igual a la dels clients, ja que el procés de facturació d'entrada és sistemàticament i legalment igual al de facturació de sortida. Per tan només necessitarem les dades de facturació al igual que per un client. I després podem afegir camps com el telèfon, o detalls o qualsevol cosa purament informativa, que en el meu cas no requerim res concret.

#### **- 3.2.5 - Taula Productes**

Aquesta taula serà la que contindrà tota la informació de totes les estadístiques, tots els estocs, productes, menús, de les imatges del explorador i informació general sobre cada producte.

Hi ha molts camps que són de lògica, com poden ser la clau primària 'id' única, el nom, la quantitat restant, el cost i el preu de venda al públic. Després tenim alguns camps d'informació general, tots opcionals, per oferir detalls sobre el producte, informació sobre la localització al local, i modificadors que se li puguin aplicar.

Referent al explorador necessitarem un camp que defineixi on està contingut aquest producte, ja que tot estarà organitzat per carpetes, amb el que necessitem un camp que identifiqui el pare de cada producte, i en el cas de ser nul o 0 entendrem

que està al directori arrel. També per l'explorador necessitem un camp de dades on emmagatzemar la imatge si es desitja tenir-ne i un camp per identificar el tipus de producte del que estem parlant, que podríem identificar-ho fàcilment per números com:

- 0 – Carpeta
- 1 – Producte en cru
- 2 – Producte de venda
- 3 – Menú

D'aquesta forma gastarem menys espai. Analitzant cadascun dels casos, tenim que el producte cru necessita un camp per les unitats, el producte de venda necessita un camp on guardar la informació dels productes en cru que conté, i el menú necessita un camp on guardar tota la informació dels productes de venda que es podran escollir per cada plat. Per tant crearé un camp 'multiproducte' on guardarem en mode text amb separadors la informació de cadascun, d'aquesta forma podrem fer un camp multi ús. I per el camp de producte cru, necessitem crear un camp 'unitats' de tipus text informatiu, per tal de conèixer com es valoraran les dades de quantitats que hi hauran dins.

Per últim ja només quedarà tenir on guardar totes les dades de les estadístiques. Com a requeriment tenim, guardar un rendiment per saber quan s'esgotarà. La única forma que tenim de fer un càlcul de rendiment és si sabem quan ha estat feta la última venda, i fer un càlcul de mitjana acumulativa amb un rendiment anterior, així tindrem una variable que anirà agafant un aprenentatge del funcionament del local. Per tant necessitem un camp per el rendiment de tipus enter ( treballarem amb segons mateix ), un camp per guardar la data dia/mes/any HH:MM:SS de la última venda, un camp per guardar el número total de ventes

efectuades, per tal de saber sobre quantes mostres de ventes està efectuada la mostra de rendiment, per tal de fer l'aprenentatge del rendiment robust i progressiu, i els dies que queden per l'esgotament ( No caldria per res ja que es podria calcular en qualsevol moment, però necessitaríem tenir una dada ràpidament a la vista sempre, per saber quan s'acabaria un producte ).

### **- 3.2.6 - Taula encàrrecs**

Aquí s'haurà de guardar tota la informació refent a un encàrrec que posteriorment serà facturat o anul·lat. Per tant necessitem segur el 'id' del usuari que ha creat el encàrrec, el 'id' del client al qui va destinat ( no requerit ja que no és obligatori), la data amb dia/mes/any HH:MM:SS, i el preu total del encàrrec. Després podríem afegir un camp de detalls, no obligatori, per informació addicional.

Per últim tenim que discutir de quina forma guardarem tota la informació corresponent als productes que s'han seleccionat de la base de dades, amb quantitat, preu, informació de menú si és el cas, i el 'id' del producte per si de cas. Es van parlar dos formes de guardar aquesta informació, primer de tot, i com és més lògic varem pensar de crear una taula nova, on posar cada encàrrec de producte, amb les quantitats i tota la informació de cada encàrrec. De tal forma que cada entrada en la taula nova corresponent seria com un encàrrec d'un usuari per complementar un encàrrec d'un client. Però això era molt extens, ocuparia massa aquella taula, i seria informació molt desorganitzada. Apart que si després hem d'utilitzar el mateix mètode per els encàrrecs facturats, seria un munt d'espai a una taula ocupat inútilment en moltes files, i convindria tenir el mateix format per els encàrrecs que per les factures. Per tant finalment hem decidit crear un camp estructurat de text on organitzarem amb separadors la informació dels productes. No

sol ser un sistema gaire recomanat, ja que sol necessitar més tasques de programació, però en el nostre cas, ja que es guardarà molta informació, ens convé per disminuir la grandària, i per agilitzar transaccions.

I en últim lloc la clau primària 'id' única, que identificarà cada encàrrec, per part nostre, i per part del client i els treballadors.

### **- 3.2.7 - Taula Factures**

Aquesta taula contindria exactament la mateixa informació que la taula dels encàrrecs, però amb alguns afegits. Necessitarem la informació de tot el tràmit de cobrament. Així com també necessitarem diferenciar si és una venda interna (per alimentació del personal) o venda externa.

Per tal de guardar tota la informació de la transacció de cobrament, necessitarem un camp que indiqui la forma de pagament, un altre per indicar la caixa on s'ha cobrat ( servirà per poder fer comprovacions extra al tancament de caixa ), i dos camps més que guardaran quans diners ha entregat el client, i quants s'han donat de canvi, per tenir una idea general de la transacció i tenir més informació.

### **- 3.2.8 - Taula de factures rebudes**

En aquesta taula volem guardar el historial de les entrades de productes a magatzem, també volem donar la possibilitat de que s'entrin altres factures no relacionades amb el magatzem, però això serà un tema totalment opcional.

Com desitgem guardar la informació de factura i el historial, necessitaré tenir on guardar el número de factura referent a la numeració del proveïdor, com no puc saber quina numeració farà servir ni si coincidirà amb d'altres, el camp serà tipus text i lliure, apart de que podria haver-hi varies entrades de diferents productes amb la

mateixa factura, i per tant necessitaré un altre camp 'id' com a clau primària, totalment independent. També farà falta un camp per identificar quin producte és el que s'està entrant, i un altre camp que identifiqui el proveïdor corresponent d'entrada, els dos enters i fen referència a les claus primàries de les taules corresponents. Altres camps necessaris són el total de la factura, l'usuari que l'introdueix, la quantitat que n'introdueix, les unitats en que l'introdueix i l'iva amb el que s'ha cobrat la factura. Així ja tindriem tot el relacionat amb la facturació d'entrada cobert.

### **- 3.2.9 - Taula extra de dades**

Aquesta taula va ser una idea d'últim moment, parlant amb el client, va sortir la idea de tenir el logo de forma centralitzada, per tal que no es tingui que configurar res de cap logo a cap dels clients, i apart tenir un lloc on si volguéssim poguéssim guardar qualsevol tipus d'arxiu. Seria com un blog d'arxius de dades. Només necessitaríem 3 camps, un camp per clau primària 'id', un camp per el nom, i un camp del tipus 'blob' ( format de dades de MySQL) per guardar les dades.

### ► 3.3 - Esquema final de l'estructura de dades

Ara ja podríem fer un esquema del resultat final de l'estructura de la base de dades. Com ja hem dit anteriorment, no ens interessa garantir una integritat de dades a nivell de gestor de dades, sinó que garantirem integritat a nivell d'aplicació, i donarem llibertat d'empleat de camps. Tot i això, afegiré les referències entre taules per saber les dependències. L'esquema el tenim a continuació a la *figura 13*.

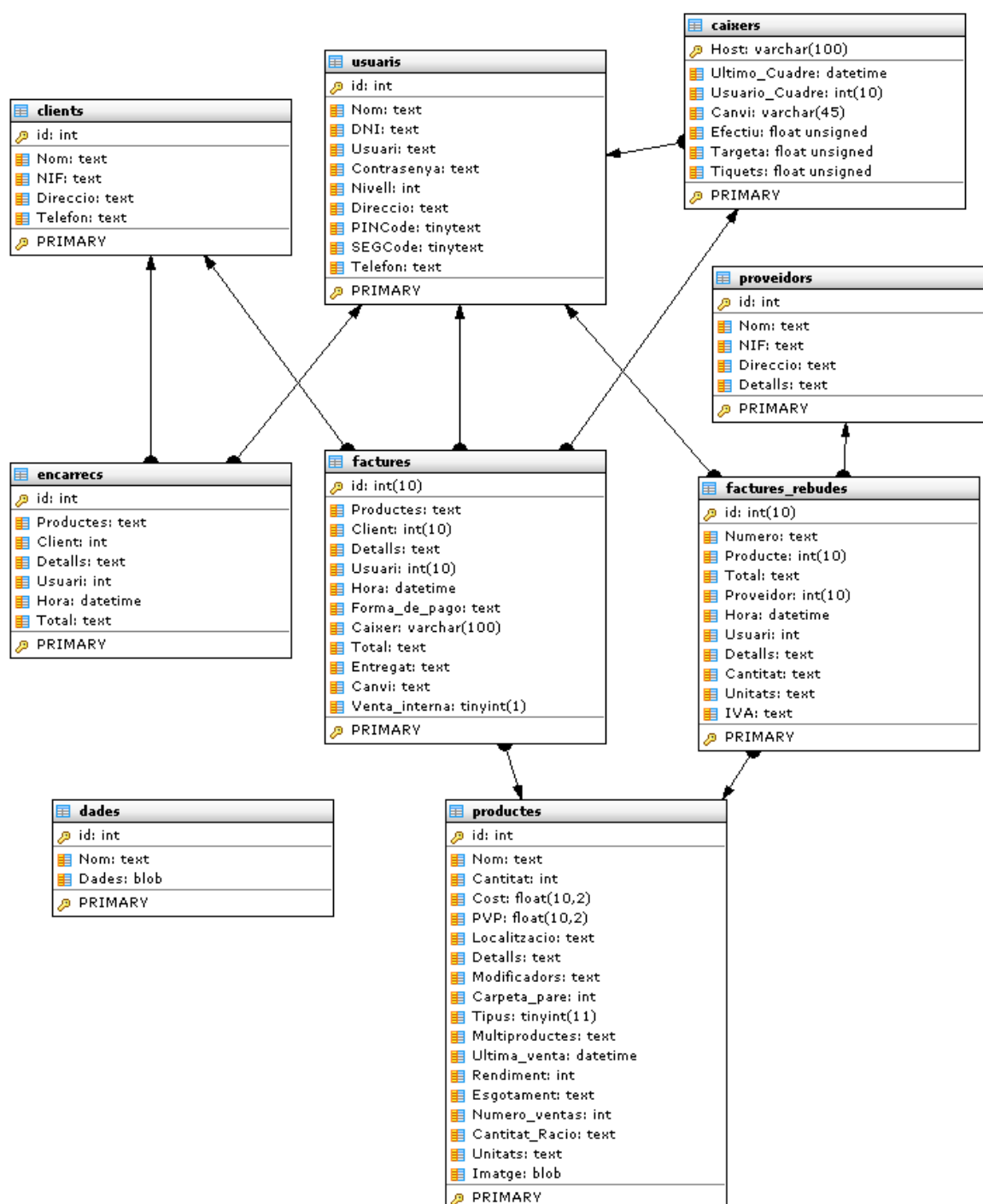


Figura 13

### ► **3.4 - Connexió amb el gestor de dades**

Per connectar una base de dades MySQL amb Visual Basic.NET, concretament amb la versió del .NET Framework 2.0, disposem de varis connectors oferts per la mateixa empresa MySQL. Existeix la versió 1.0, versió molt funcional i estable però amb limitacions de funcions sobre ADO.NET 2.0 i també sobre versions 5.0 o superiors del servidor MySQL ( Treballa normalment amb versions 4.x ). Existeix la 5.1, la més avançada de totes i amb més funcions, però actualment no es troba en estat funcional i estable, ja que està en fase de proves. I la última que queda és la 5.0, la més equilibrada, i la que millor ens funcionarà, ja que les funcions extres des de la 5.0 a la 5.1 tampoc tenim cap necessitat d'utilitzar-les. Apart de tot això també existeix un pluguin per el Visual Studio, però tampoc ens interessa perquè està més destinat a treballar en forma de formularis, i nosaltres volem una aplicació molt feta a mida. En general només necessitem un entorn per executar consultes i instruccions, que suporti transaccions, transferència de dades en cru ( pujar i baixar imatges o arxius ), i que sigui fiable i estable.

**Referència 3.**

#### **- 3.4.1 - Creació i manteniment de la connexió**

Examinant el connector, veiem que no ens ofereix la comoditat que desitgem, ja que si tenim que està fent consultes gaire sovint no ens interessa tenir que fer totes les crides, ni tornar a crear la connexió, ni anar passant variables amunt i avall. Apart que el control i recuperació d'errors seria molt limitat i dificultós, i hauríem d'escriure molt codi. Per això vaig decidir crear una nova classe, que contindrà una connexió del tipus MySqlConnection de la branca MySQL.Data.MySQLClient de la llibreria MySQL.Data.dll per tal de poder crear tantes connexions com vulguem, i que

cadascuna d'elles ( totes del tipus de la nostre classe ), tingui el seu propi control i gestió. I així doncs minimitzar tot el codi necessari des del exterior de la classe per fer les funcions que haguem de necessitar fora, i englobar-ho tot el millor possible dins de la classe.

Les coses que necessitem en aquesta classe són totes les que posteriorment haurem de fer servir, per tant he de tenir molta visió per donar-li bona forma. Sabem del tot segur que necessitarem un gestor de la connexió, amb un formulari afegit per configurar-la ( encara que només sigui per la primera execució, ens interessa que ho pugui fer l'usuari final ), i provar les configuracions que se'ns introdueixin, i fins que no funcioni no tirar endavant. Per tant requerirem d'un bucle de connexió, que vagi provant, i un control d'error sobre el intent de connexió, per saber si la connexió ha fallat o no, i controlar-ho, per poder mostrar el formulari de configuració si és necessari. La configuració de la connexió la podem agafar directament del constructor de la classe ( la funció New definida amb tots els arguments necessaris per la connexió ), o també ho podrem fer posteriorment amb una funció de canvi. La connexió simplement es fa amb la funció Open, de la classe MySqlConnection, i la gestió d'errors en Visual Basic.NET funciona amb les expressions Try....Catch ex as exception....End Try. Per tant, introduïm totes les dades a la connexió, cridem el bucle de connexió, i provem dins del control d'errors si connecta, en cas de connexió la funció acaba i la classe queda inicialitzada i connectada, en cas negatiu, es crida el formulari de configuració, l'usuari pot canviar, i es va provant.

Ara la connexió ja es crea, i podríem connectar, però aquestes dades s'han de guardar a algun lloc, per tal de que el pròxim cop que arrenquis el software no tinguis que introduir res. I hem de tenir molta cura d'aquestes dades, ja que són la localització, l'usuari i la contrasenya de on es guarda tota la informació del negoci en

qüestió, cosa que no pot caure en males mans. Així doncs hem de gravar-ho al registre, però haurem de gravar-ho ben codificat perquè no es pugui llegir ni robar fora de la nostre aplicació.

Fen una ullada ràpida a la documentació del Visual Basic.NET [4], ràpidament trobaré dos llibreries que hem seran molt útils per aquesta situació. La primera és la Microsoft.Win32 que conté la classe RegistryKey, la qual ens dona suport per treballar amb el registre de Windows, per tal de guardar-hi i extreure'n informació. Però necessitem una forma de codificar la informació que guardem, perquè no la puguin extreure. També molt ràpidament es pot trobar a la documentació, una altre llibreria anomenada System.Security.Cryptography la qual conté classes com la TripleDesCryptoServiceProvider per codificar objectes, la MD5CryptoServiceProvider amb la que generarem el hash de la clau secreta que utilitzem per codificar amb el sistema que desitgem. Fàcilment buscant per la documentació es troba cadascuna de les funcions, les variables, els tipus, i com funcionen cadascuna de les parts de les funcions, per tal de codificar de forma segura un text. Tot es fa en binari, passant tota la informació a bytes, es codifica amb dos Streams, CryptoStream on es genera la codificació i MemoryStream que utilitzarem després per extreure el codi binary i donar-li una forma en text per poder-ho guardar correctament al registre ( Ho farem en hexadecimal mateix ). Tot estarà amb un control d'errors per si de cas, i després en la descodificació només es tracta de fer el procés al revés.

### **- 3.4.2 - Funcions d'entrada i sortida**

Doncs ara ja podem connectar a la base de dades, podem guardar la informació de forma segura, i podem controlar els errors, ara necessitem crear les

funcions de consultes e instruccions necessàries. Les funcions més importants que hem de crear són les de entrada i sortida des de l'aplicació a la base de dades. Les anomenaré 'consulta' i 'executa'. La funció 'consulta' serà una de les més executades per l'aplicació, per tant haig de mirar de que sigui senzilla i ràpida. Això hem porta primer de tot a decidir que no inclourà res relacionat amb extracció d'arxius o imatges de la base de dades, ja ho faré en una altre per quan sigui necessari.

#### **- 3.4.2.1 - Funció consulta**

Una consulta ha de ser ràpida, per tant faré servir el mínim número d'instruccions per aconseguir un resultat, i mirar d'oferir-lo en un format còmode. La llibreria de MySQL porta dins la funció MySqlCommand que serveix per generar la comanda consulta, i si ho requerim indicar-li la transacció amb la que estem treballant, i després també tenim la funcio MySQLDataAdapter, que ens retornarà tota la informació de la comanda en un format DataTable, molt comú i fàcil d'utilitzar en Visual Basic.NET (Es tracta d'una classe casi com una matriu però amb algunes opcions de format i funcions de gestió). Tota la funció l'acompanyarem d'un control d'error, i en cas d'error cridarem directament el bucle de connexió, que ja s'ocuparà de tornar a connectar i demanar a l'usuari la informació necessària, i així un cop funcioni tot, donar com a sortida de la funció una crida recursiva a la mateixa funció amb els mateixos paràmetres, tot sempre interrompible per l'usuari i amb pantalles de notificació del que va passant. Només cal afegir alguna comprovació de si la connexió està activa, per no perdre temps donant errors, si ho podem saber directament.

### **- 3.4.2.2 - Funcions de consulta d'imatges**

Com de moment només necessitem poder extreure les imatges de cadascun dels productes, i la imatge del logotip, faré dos consultes directes que retornin en format 'Image' de Visual Basic.NET cadascun d'ells, sempre només en el cas de que l'usuari desitgi utilitzar l'aplicació amb imatges, que serà una variable booleana que hauran de poder modificar des del panell de configuració de la connexió, i s'haurà de guardar al registre, de la mateixa forma que guardem les dades de connexió però sense codificar.

Per la funció de la imatge de cada producte podrem fer una crida directament amb la 'id' del producte que desitem la imatge, la consulta SQL tindrà un format fixa sobre la taula de productes segons la 'id' que demanem, i tot funcionarà igual que a la funció de consulta fins al moment de fer servir el MySQLDataAdapter, ja que ara no requerim una DataTable, sino que requerim llegir dades, per tant necessitem la classe MySQLDataReader, la qual rebrà la sortida de la funcio ExecuteReader de la classe MySqlCommand, i ens retornarà el camp que desitem en format binari, que directament amb la llibreria System.Drawing que conté la classe 'image', podem fer una crida de la funció FromStream, i entrar-li un MemoryStream que contingui la imatge en el format binari que ja tenim.

La funció de consultar el logotip, serà igual que l'anterior canviant únicament la comanda SQL que executarà el MySqlCommand. Sabem que el logotip es guardarà a la taula de dades, per tant podem fer una consulta segons el 'nom' a la taula, o simplement agafar el MAX(id) de la taula per saber quin és l'últim logotip introduït. El qual també l'introduïrem des del formulari de configuració de la connexió.

### **- 3.4.2.3 - Funció executa**

Com la funció 'executa' no la farem servir tant com la de consultes, si que li introduiré com a opcionals les opcions d'introduir arxius. En un principi aquesta funció varem decidir que fes una execució sobre ella mateixa, gravant tota la informació de quan, com i qui, a fet aquella execució (Ja pot ser UPDATE, INSERT o DELETE), però posteriorment ho vaig trobar una mica absurd, ja que ompliria molt espai, requeriria una taula més, i el servidor MySQL ja te un historial propi, que seguint-lo una mica es pot trobar tota la informació en cas de problemes.

Aquesta funció segueix la mateixa estructura que les consultes, el mateix control d'errors, i les mateixes comprovacions, però és encara molt més senzilla. Farem servir també la classe MySqlCommand, la qual en cas d'inserció d'imatge haurem d'afegir un camp '?File' a la consulta SQL, fent referència a un paràmetre que es passarà com a argument a la classe de la comanda, i serà l'arxiu que desitgem enviar en binari. Un cop afegit el paràmetre a la comanda amb la funció '.Parameters.Add(...)', com no tenim que retornar res, simplement cridant la funció ExecuteNonQuery de la classe de la comanda ja haurem acabat.

### **- 3.4.2.4 - Funcions d'informació**

Aquestes funcions són totalment opcionals, però ja que la classe MySqlConnection estarà continguda dins de la meva classe, probablement en algun moment desitjarem poder consultar-ne informació còmodament. Per tant crearem funcions amb noms propis meus, per cridar a funcions que ens retornaran el nom de la base de dades, la versió, la configuració, l'estat de la connexió i una última per si desitgem canviar de base de dades sense canviar de host ni tancar la connexió. Així doncs ja hem podria connectar introduint les dades en temps de compilació.

## 4 - Interfície Gràfica d'usuari

Al desenvolupar la interfície gràfica per l'usuari també hauré d'explicar moltes funcions que no són gràfiques, i moltes rutines que es duen a terme en segon pla perquè tot funcioni be. Sempre fent explicacions superficials i donant per suposat la majoria dels temes tècnics de com crear les rutines, i evitant comentar o argumentar codi.

Basaré molt el desenvolupament en els formularis, ja que són les classes principals que diferenciaren l'esquema funcional, i les necessitats i requeriments en cada moment.

Cada formulari necessita funcions d'inicialització, funcions per cada botó, i pot contenir funcions per cada succés de qualsevol objecte de d'interfície. En principi explicaré només les que són més relatives al meu projecte, les que s'encarreguen d'acomplir els requeriments, totes les que són molt freqüents en el desenvolupament en Visual Basic.NET, per fer servir objectes comuns com els formularis (Tancar, obrir, etc...), botons, quadres de text, quadres de selecció, etc... rarament en comentaré res.

## ► 4.1 - Formulari de configuració

Ara que ja tenim tot el sistema intern de connexió muntat, tenim que donar-li forma al formulari corresponent a la configuració. Haurà de contenir el servidor, l'usuari, la contrasenya, la base de dades, la possibilitat de veure o canviar el logotip i poder seleccionar si volen treballar amb imatges o no. El procés de modelat de la interfície és molt senzill i no requereix cap explicació, però perquè tingueu una idea de l'aspecte de l'aplicació aniré adjuntant imatges. El formulari de la configuració tindrà l'aspecte següent :

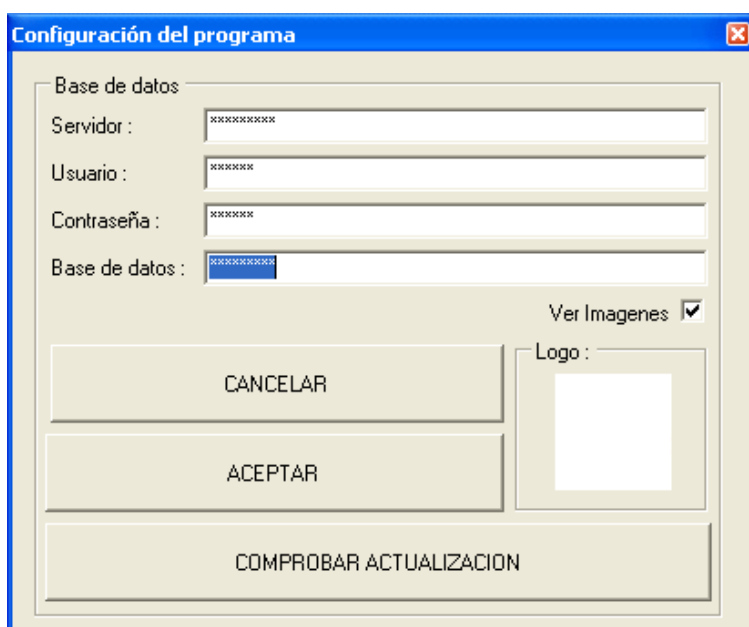


Figura 14

- El sistema d'actualització automàtica està en fase de proves, no és requeriment per el projecte -

Com podem observar a la figura 14 els camps de configuració estan emmascarats com contrasenyes, això és per si de cas algun client estigues per allà en el moment d'encendre algun ordinador nou, o en el cas de que ocorregués algun problema, les dades sempre estarien preservades i codificades a qualsevol mirada curiosa.

Aquest formulari haurà de contenir totes les funcions referents els successos d'interfície al prémer cada botó, o al prémer el logotip. Al prémer el logotip s'obrirà un

OpenFileDialog (Que és una classe comú a Visual Basic.NET) per buscar la imatge a enviar cap a la base de dades, i de moment guardar la informació a la mateixa imatge que es visualitza, per després al prémer acceptar, gravar-ho tot.

Necessitaré una funció per inicialitzar tot el formulari, aquesta funció haurà de saber si disposem de connexió per carregar el logotip o no, i haurà de buscar totes les dades del formulari al registre i emplenar els requadres, en cas de que existeixin.

I per últim, al prémer el botó d'anular, simplement que amagui el formulari, i si es prem el d'acceptar, gravaré totes les dades al registre amb la classe RegistryKey, i després transformaré la imatge del logotip en un MemoryStream en format binari, i l'enviaré a la base de dades amb la funció 'executa' i el paràmetre corresponent, tal i com he explicat anteriorment.

## ► **4.2 - Mòdul de suport**

Aquest mòdul el tractarem com una classe publica per tothom que contindrà en arrel del software totes les funcions bàsiques de suport per l'aplicació, i totes les variables que necessitaré com a globals per treballar còmodament durant tot el desenvolupament del projecte. Per exemple podríem ja incloure aquí les funcions de codificar i decodificar text per si les volem utilitzar en alguna altre part del software.

També hi inclouré moltes funcions que s'utilitzaran al llarg del desenvolupament, i que explicaré més tard. Algunes d'elles es podria considerar opcional incloure-les en aquest mòdul, per tant, com és només un tema d'organització superficial, només explicaré les que si seria obligatori d'incloure.

### **- 4.2.1 - Variables globals**

Com a variables globals tindrè la classe de connexió que acabem de crear, la configuració de la connexió en qüestió per si de cas, el valor d'iva de facturació, la variable booleana referent a si volen imatges o no, la 'id' del usuari que hagi entrat, el seu nom, el seu nivell, el tipus de coma que utilitza per els decimals el sistema operatiu i la base de dades, i cadascun dels formularis que anirem creant, tots inicialitzats en buffer per tal de donar un funcionament més ràpid al saltar d'un lloc a un altre de l'aplicació.

### **- 4.2.2 - Detecció de separador decimal**

Les primeres funcions que introduiré són les de detecció del tipus de coma decimal del sistema operatiu i la base de dades. La de la base de dades com ja sabem que configurarem el MySQL en format estàndard, ja sabem que vindrà amb 'punt' (Format anglès), però per el sistema operatiu, haurem de provar una conversió

de dades d'un nombre decimal, per exemple amb 'punt' i veure si el retalla, o l'agafa complert, si el retalla tindrem que funciona amb 'coma', sinó, amb 'punt'.

#### **- 4.2.3 - Format fixa dels nombres decimals**

Una altre funció important a introduir és la encarregada de passar qualsevol número decimal a 2 xifres decimals, i transformar-lo en text amb la coma corresponent a si va destinat a sistema operatiu o a base de dades. Constarà de dos parts, una en la que agafem el nombre i li retallem els decimals, i una altre en la que li donarem format al nombre com a text, on sempre mostrarem els 2 decimals encara que siguin zeros, ja que volem tota la informació sobre els diners en pantalla, i volem un format fixa. Per retallar els decimals serà simplement una funció matemàtica per arrodonir de forma legal de 0,5 amunt i inferior avall. I per el format com a text, primer haurem de separar si és per base de dades o sistema operatiu, passar la variable a text amb la coma corresponent i comprovar els decimals que té per afegir els 0 corresponents i donar un format fixa.

#### **- 4.2.4 - Carrega de configuració**

També serà obligatori per començar a treballar la funció que carregarà les dades de configuració del registre, per tal de poder inicialitzar la classe de la connexió directament al executar l'aplicació. Hauré de treballar amb les funcions que ofereix la classe RegistryKey, per llegir cadascuna de les dades i donar-li els valors a les variables globals corresponents.

#### **- 4.2.5 - Sortida per Visor VFD Sèrie**

Aquí podria incloure ja totes les funcions per mostrar text per els Visors VFD externs connectats per COM sèrie. No caldrà cap llibreria externa ja que els visors són genèrics i funcionen amb seqüències de caràcters directament enviades en mode ASCII per el port sèrie. Per tant crearé dos funcions, una per mostrar pel Visor el text desitjat, escollint a quina línia es vol mostrar (Tenen 2 línees de 20 caràcters), i una altre que cridarà la funció anterior per mostrar la informació d'un producte en concret, ja que serà molt freqüent mentre es faci funcionar l'aplicació.

### ► **4.3 - Formulari principal (Part 1)**

Aquí ja he de començar a donar-li forma a l'aplicació, per tant haig de crear una interfície que doni accés a totes les parts del programa, i un control temporal a la mateixa interfície per comprovar la inserció/extracció de targetes chip per ingressar.

L'aspecte visual l'hem de crear perquè es pugui fer anar còmodament amb monitor tàctil, i ha de tenir tots els accessos directes als altres formularis. Ens quedaria com a la *figura 15*.



*Figura 15*

- L'apartat de presencia està en fase de proves, no és requeriment per el projecte -

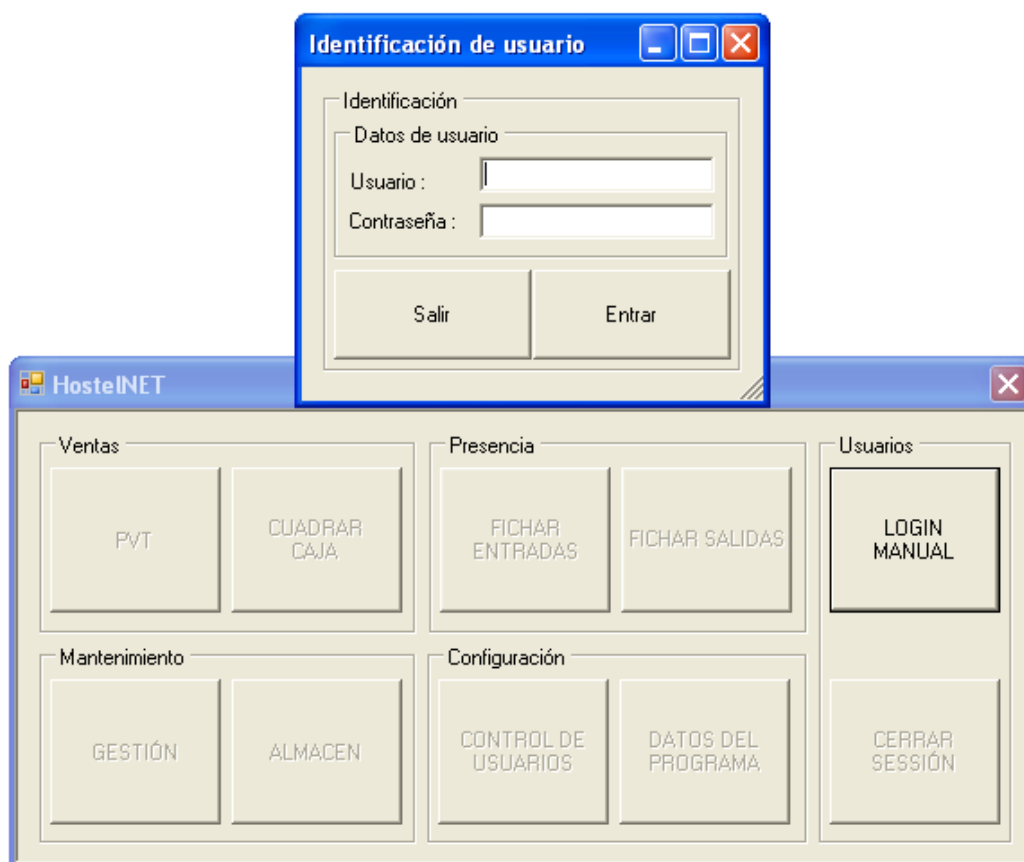
Es poden observar els botons grans per el monitor tàctil (Mida còmode per un dit), i tots els accessos bloquejats ja que encara no s'ha ingressat com a cap usuari. Haurem de mirar de que només accedir a aquest formulari (El primer que surt al arrancar), s'inicialitzi la connexió (En cas de problemes la classe de la connexió ja demanaria la informació).

De moment no tenim cap restricció d'usuaris, però ens interessa crear una forma d'ingressar, per tant faré que el software per si sol, si troba la base de dades d'usuaris buida, generi un usuari SYSTEM amb accés a totes les opcions

(Denominaré accés a tot com nivell 10, ja que es diferenciarien per nivells els usuaris).

## ► **4.4 – Formulari d'ingrés manual**

L'usuari que haurà generat el software automàticament serà l'usuari SYSTEM amb contrasenya SYSTEM de nivell 10, i amb totes les altres dades buides. Així ja podrem crear i provar el sistema d'ingrés. Per tant restringiré l'accés a totes les opcions del formulari principal menys a la d'ingressar, li donaré forma al formulari d'ingressar, i enllaçaré el botó corresponent. A la *figura 16* es pot observar.



*Figura 16*

Aquest formulari d'ingrés podria comprovar si estem treballant amb una base de dades correcte (Amb totes les taules i camps necessaris) però això és opcional, ja que es dona per suposat que la base de dades serà la correcte. Principalment haurà de fer una consulta a la base de dades comprovant si existeix algun usuari amb els usuari i contrasenya introduïts. En cas afirmatiu haurà de mirar de quin nivell

és l'usuari, restringir l'accés a les opcions segons sigui necessari i donar els valors necessaris a les variables globals referents a l'usuari.

Els nivells els diferenciaré de moment per càrrecs, un usuari de nivell 1 seria un treballador de venda, només amb els botons de TPV i quadre de caixa, que són els únics que necessita. El de nivell 2 seria un encarregat de magatzem, amb accés a nivell 1 i afegint el botó de magatzem. El nivell 4 el posaré per encarregat de personal, també amb accés als nivells inferiors i afegint el botó de control d'usuaris. I per últim el nivell 9 o superior amb accés a tot, per tant, tots els nivells anteriors i afegint el botó de la configuració del programa i la gestió econòmica. Només afegir, que un cop ingressat, també s'activarà per tots el botó de tancament de sessió, que tornarà a deixar-ho tot restringit i esborrarà les dades d'ingrés anteriors.

## ► 4.5 - Formulari de control d'usuaris

Com ja tenim el formulari principal, i el software pot arrencar i crear la connexió correctament amb la base de dades, també tenim un usuari SYSTEM i podem ingressar manualment, ja puc començar a crear els formularis de treball.

Per tant començaré a treballar amb la base de dades d'usuaris. Necessitaré un formulari per poder crear, modificar i esborrar usuaris, amb tota la seva corresponent informació. I que també doni la possibilitat de generar les targetes per cadascun d'ells. Primer li donaré l'aspecte visual, a la *figura 17* es pot observar.

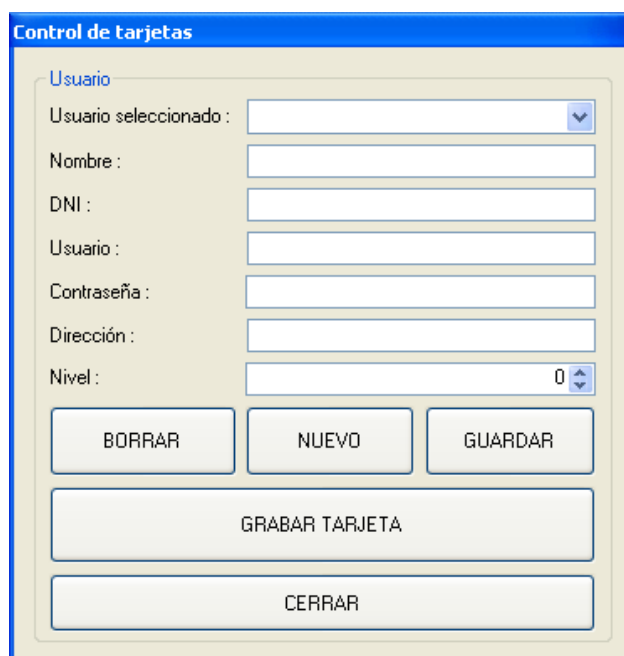


Figura 17

Enllaçaré directament el botó de control d'usuaris del formulari principal amb aquest formulari, i ja podrem entrar a fer proves amb l'usuari SYSTEM.

Quan el formulari inicialitzi, llegirà tots els usuaris de la base de dades amb nivell inferior al usuari ingressat (I també l'usuari ingressat), i els mostrarà a la barra superior (*Figura 18*).

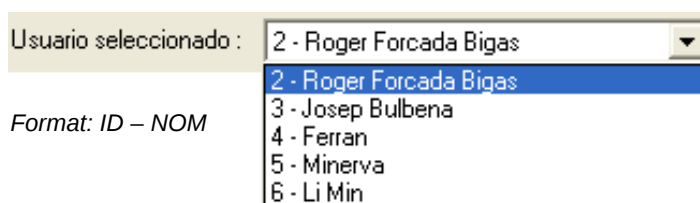


Figura 18

- Un cop ja tinguem usuaris creats farem l'usuari SYSTEM secret -

Si seleccionem un usuari carregarà totes les dades als quadres de text, i utilitzarem una variable global 'idse' per guardar la 'id' del usuari que tenim seleccionat. La Mateixa variable global la farem servir per identificar si estem modificant o creant un usuari, ja que en cas d'estar creant un usuari nou seria 0. Per tant el botó de nou usuari només haurà de buidar tots els quadres de text i posar la variable 'idse' a 0. Però així encara no genero l'usuari, ni el modifico, falta el botó de guardar.

El botó de guardar haurà de mirar si estem modificant o creant usuari i fer la corresponent instrucció SQL amb UPDATE o INSERT i els camps dels quadres de text.

El botó de esborrar haurà de mirar si tenim algun usuari seleccionat, mirar que no siguem nosaltres mateixos, preguntar si realment el desitges esborrar i fer la corresponent instrucció SQL amb DELETE.

#### **- 4.5.1 - Gravació de targetes**

Seguint amb el formulari d'usuaris, ara necessitem generar les targetes amb el botó de gravar targeta. Com ja he comentat en els requeriments, les targetes han de ser úniques, no s'han de poder copiar, i la informació ha d'estar codificada. Tenim els dos camps a la taula per el codi PIN, i un per el codi de seguretat que identificarà cada usuari. Per facilitar la generació d'aquests dos codis, fent una ullada a la documentació del .NET Framework 2.0, trobaré moltes funcions destinades a generar números aleatoris, però no seqüències de lletres. Per tant crearé una classe nova, una mica estructurada per facilitar-me les coses, que a partir d'un bucle de generacions de números hem retorni seqüències de lletres aleatòries.

Abans de poder fer res amb cap targeta, s'haurà de configurar i provar tot el sistema de targetes, ja que els lectors/gravadors de targetes chip, encara que siguin genèrics, no són directament suportats per cap llenguatge, i requereixen de llibreries externes genèriques ( API's de Windows). Per tant haurem de fer crides a funcions d'una llibreria DLL externa. La llibreria s'anomena SCARD32, i només necessitaré la funció SCardCommand, que funciona passant com a argument de text la comanda que pretens executar, i retorna un objecte de tipus variable, contenint el que torna directe de la API aplicant la comanda corresponent al lector/gravador.

Al gravar una targeta nova per un usuari, l'aplicació comprovarà el pin guardat a la base de dades amb la targeta introduïda, en cas de no ser correcte, demanarà introduir el codi correcte (Les targetes venen amb un codi PIN estàndard i els usuaris es creen sense cap PIN a la base de dades). Un cop tinguem el PIN correcte, en generarem aleatòriament un de nou, i també generarem un nou codi de seguretat. A la targeta i gravarem la seqüència (id usuari):HostalNET:(Codi de seguretat nou), i li canviarem el PIN per el generat. I un cop gravats el nou PIN i el nou codi de seguretat a la base de dades, ja tindrem la targeta creada i totes les dades bé.

## ► **4.6 - Formulari principal (Part 2)**

Ja podem iniciar sessió, crear usuaris, modificar-los, esborrar-los i crear les targetes úniques per cadascun d'ells. Ara falta crear l'apartat de lectura de targetes, per tal de que també puguin ingressar amb la targeta chip. Això ho farem en el formulari principal, ja que serà el primer formulari que sortirà al obrir l'aplicació, i juntament amb el d'ingrés manual, seran els únics llocs on es podrà anar sinó s'ha ingressat com usuari.

### **- 4.6.1 - Lectura de targetes**

Crearé un temporitzador, que cridi un succés cada un determinat interval de temps. Aquest succés cridarà una funció que de forma molt rapida, anirà comprovant per nivells totes les possibilitats i seguretats de la targeta.

Primer mirarà si hi ha targeta en el lector (en el cas de no tenir ni lector, la llibreria retorna com si no hi hagués cap targeta, cosa que va perfecte), després miraré si la targeta és del tipus que a mi m'interessa. Després si tot es correcte, en llegiré el contingut, i miraré que tingui el format que se li dona quan es graven. Si te el format correcte, miraré si el codi de seguretat és el correcte per la 'id' de usuari gravada a la targeta, i en el cas de que també sigui correcte, verificaré si el PIN de la targeta és el mateix que està guardat a la base de dades. Un cop totes les comprovacions són afirmatives, cridarà a la mateixa funció d'ingrés que utilitzo un cop la comprovació d'ingrés manual dona usuari i contrasenya correctes. Així que els usuaris ja podran ingressar amb targeta.

## ► **4.7 - Formulari de control d'estocs**

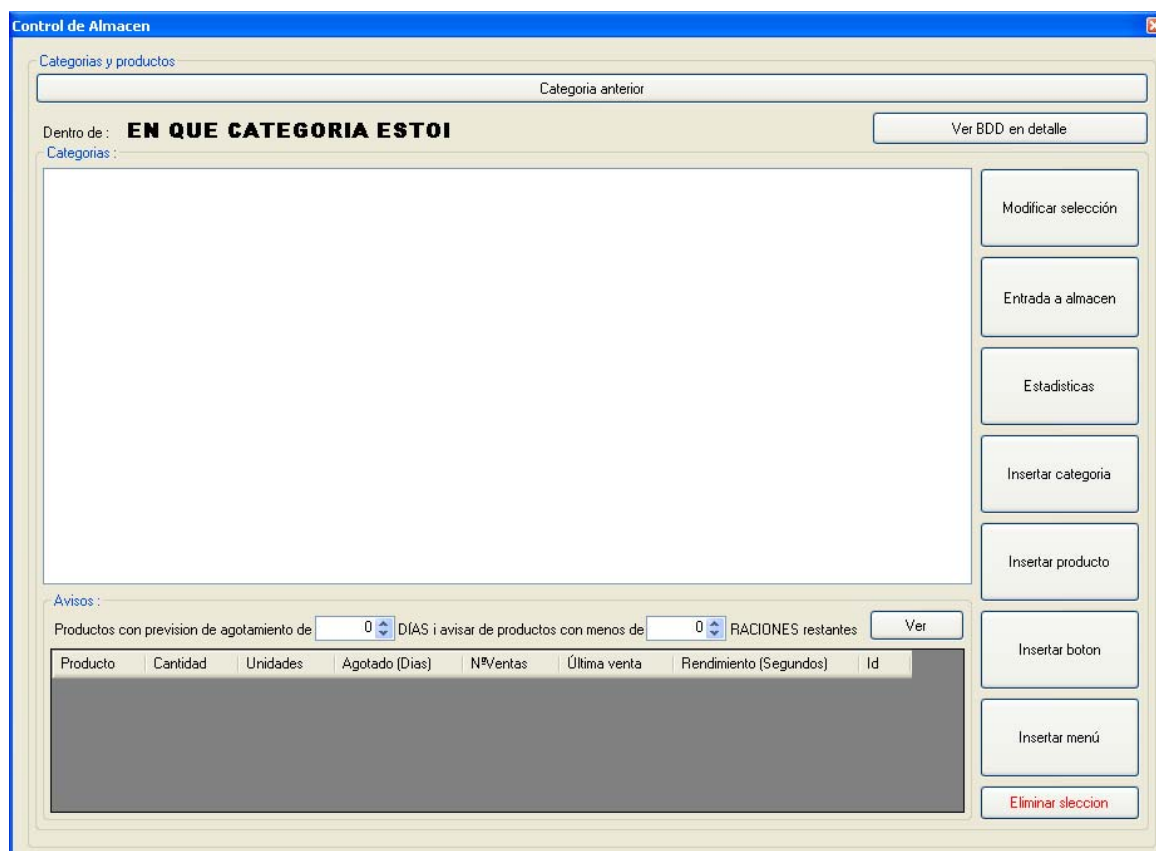
Aquest formulari serà molt extens, i tindrà molts subformularis, per tant és molt important pensar bé tots els requeriments que s'inclouen en ell. Per començar necessitem poder veure i buscar entre tot el magatzem (Tots els productes siguin del tipus que siguin) còmodament, tenint en compte que s'han d'organitzar per carpetes.

Es té que poder crear, esborrar i modificar qualsevol tipus de producte, i carpetes (Seran productes nuls, amb imatge si es desitja), amb totes les opcions que ha de tenir cadascun segons el seu tipus. S'ha de poder donar entrada a qualsevol producte cru. Tenim que poder veure les estadístiques detallades d'un producte en concret, i veure un resum d'avisos d'esgotament i rendiments dels productes que els i quedi poca quantitat, o pocs dies per esgotar-se.

Tots els càlculs de rendiments i esgotaments els explicaré més tard, ja que ara no hem calen. Els posaré tots en el mòdul base per poder-los utilitzar des del magatzem i des el TPV.

Com ja sabem tot el que ha d'incloure, per poder fer proves còmodament de les funcions que aniré generant, primer de tot li donaré un aspecte visual com en tots els altres casos. Haurà d'incloure l'explorador visual amb imatges, per navegar per productes i carpetes, botons per afegir cadascun dels tipus de producte, un botó per eliminar el producte seleccionat, un per modificar-lo (La detecció de tipus és automàtica), un per mostrar-ne les estadístiques i un altre per donar-ne entrada en el cas de que sigui producte cru. Alguns d'aquests botons, com el d'esborrar, no necessiten explicació. Després posaré la opció de veure tota la base de dades dels productes en format taula, per buscar més tècnicament. I per últim faltará l'apartat d'avisos d'esgotament. L'usuari podrà seleccionar el llindar de dies que falten per esgotar-se o el llindar d'unitats restants dels productes a mostrar als avisos, i els

avisos els organitzaré en una taula amb totes les dades necessàries. A la *figura 19* ho podeu observar com queda.



*Figura 19*

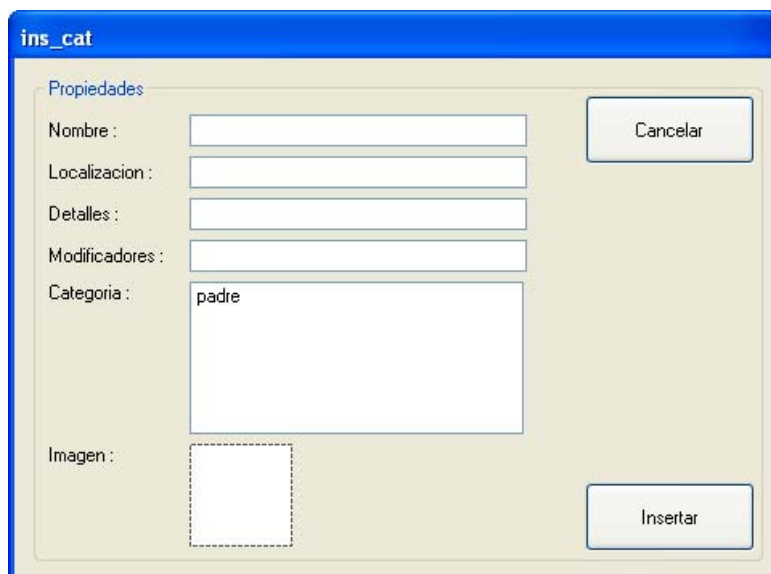
La selecció desitjada de dies o unitats restants la guardaré i carregaré del registre del Windows, amb la classe RegistryKey, per tal de que no es tingui que anar modificant sempre. Per tal de fer la mostra a la taula inferior, només s'ha de fer la crida a la base de dades de la consulta SQL corresponent indicant les clàusules de dies i esgotament inferiors a lo desitjat, i com ja ve en el format DataTable que és el format del DataGridView.Source (Origen de dades de la classe), només caldria organitzar les dades a la consulta SQL de la forma que s'han de mostrar després i ja estaria (També hi ha la possibilitat de recórrer la DataTable com a matriu i anar afegint línees a la DataGridView).

Ja tenim el formulari visualment creat, però abans de poder mostrar cap producte, modificar-lo ho esborrar-lo, he de saber com els crearé, i quin format tindran. Per tant explicaré la creació i format de cadascun dels productes per separat, conjuntament amb el seu formulari de creació, que ja estarà enllaçat amb el botó corresponent.

L'explorador haurà de tenir una variable global de la carpeta en la que es troba, i una variable local del producte que té seleccionat. Així quan cridem a esborrar un producte només haurem de mirar aquestes variables. I quan cridem a modificar-lo o crear-lo, farem que tots els formularis corresponents s'inicialitzin amb la variable de la carpeta en que ens trobem i la 'id' del producte a modificar, que en el cas de crear nou, serà una 'id' igual a 0, al igual que en el formulari de control d'usuaris.

#### **- 4.7.1 - Creació de carpetes**

Per les carpetes no ens interessa absolutament casi cap dels camps de productes, tots podrien ser opcionals menys el nom i el camp que indica on està ubicada (El camp Pare, que indica la 'id' de la carpeta on està contingut un producte, o 0 si està a l'arrel). Però donaré la possibilitat opcional d'omplir els camps relacionats amb informació general, ja que també poden informar sobre tota una categoria sencera, i donaré la possibilitat d'assignar-li una imatge. L'aspecte de l'interfície es mostra a la *figura 20*.

The image shows a Windows-style dialog box titled 'ins\_cat'. It has a blue title bar and a light beige background. On the left, under the heading 'Propiedades', there are five labels with corresponding input fields: 'Nombre:', 'Localizacion:', 'Detalles:', 'Modificadores:', and 'Categoria:'. The 'Categoria:' field contains the text 'padre' and has a larger text area below it. At the bottom left, there is an 'Imagen:' label next to a small square placeholder with a dashed border. On the right side of the dialog, there are two buttons: 'Cancelar' at the top and 'Insertar' at the bottom.

*Figura 20*

Si estem modificant una carpeta, el formulari ja serà cridat a inicialitzar des del formulari del magatzem amb la 'id' corresponent, carregant així totes les dades als quadres de text. En el cas d'estar creant (inicialitzat amb 'id' igual a 0), en el camp de la categoria o producte pare, que contindrà totes les carpetes existents a la base de dades i una primera línia d'arrel igual 0, hauré de seleccionar el producte 'pare' que ens donarà el formulari magatzem, corresponent a l'estat de l'explorador. Així ja crearem els productes on estiguem, i encara tindrem l'opció d'escollir on aniran a parar. Així doncs només quedarà la imatge que es carregarà de la mateixa forma que el logotip al formulari de gestió de connexió. Després de totes les inicialitzacions, l'usuari podrà modificar el que vulgui, i al prémer el botó d'acceptar, es farà la comanda SQL UPDATE o INSERT, segons modificació o creació.

#### **- 4.7.2 - Creació de productes crus**

El funcionament de la interfície serà el mateix que en el cas de les carpetes, en tots els sentits. L'únic que diferencia uns productes dels altres són els camps de dades que fan servir, i en alguns casos l'estructura d'aquests mateixos camps.

Per tant faré servir tots els camps anteriors, més el camp de les unitats del producte. I també donaré la possibilitat, només en el cas d'estar modificant un producte cru ja existent, de poder canviar la quantitat, la quantitat per ració, el cost i el preu de venda. Però en principi aquests últims camps serien fixes i bloquejats. Ja que ens interessa una quantitat només relacionada amb les entrades a magatzem que haguem fet. La quantitat per ració igual a 1 per poder fer els càlculs bé. El preu de venda i el cost han de ser calculats automàticament, el cost ha de ser promig de costos d'entrades del producte (A nivell de facturació), i el preu de venda fen el promig de preu de venda de tots els altres tipus de productes que l'utilitzin.

Per tant amb tots els camps de la forma que hem comentat anteriorment, es pot observar com queda a la *figura 21*.

Insertar producto

Propiedades

Nombre :

Localización :

Detalles :

Modificadores :

Categoría : padre

Cantidad :

Cantidad / Ración :

Unidades :

Coste :

PVP(aprox) :

Imagen :

Cancelar

Insertar

*Figura 21*

Simplement quadrarem les dades noves al carregar el formulari (Segons modificació o creació), i a la hora de guardar a les comandes SQL, i ja ho tindrè tot fet.

### **- 4.7.3 - Creació de productes de venda**

Els productes de venda, tal i com explico en els requeriments, són els botons de venda bàsics. Per tan hauran de tenir totes les dades d'informació general com les de la carpeta, més la de preu de venda, i alguna forma de seleccionar quins productes inclourà. Primer de tot he de saber com organitzaré i guardaré tota la llista de productes, amb quantitats i tipus a la base de dades. Tenim el camp 'multiproducte' per fer-lo servir per les necessitats de cada producte en concret, i havia dit de fer-ho tot amb separadors de text. Per tant faré servir 2 separadors de text, un per separar el 'id' del producte i la quantitat que se'n agafa a cada venda, i un altre per separar els diferents productes entre ells. Ara necessitaré donar-li una forma visual per escollir còmodament quins productes volem incloure en aquest botó de venda. Per tant crearé una taula de 3 columnes a la interfície, on mostraré tots els productes crus possibles, amb el nom, les unitats i una columna que indicarà, per cada fila, quan se'n agafa d'aquell producte. Llavors només caldrà fer una funció, que a cada modificació d'aquesta taula, revisi totes les línees, en busca de totes les que no siguin 0, i les concateni amb les dades i els separadors corresponents en alguna variable global. Un cop actualitzada la variable global de tots els productes que es voldran vendre en el pac, també estaria bé crear indicadors visuals amb la suma del cost de tot el què s'inclou, i la suma del preu de venda. I per últim, varem discutir donar la possibilitat de limitar les quantitats dels productes de venda (Com per fer promocions limitades), però no és requeriment i de moment no ens interessa. La interfície quedaria com a la *figura 22*.

| Producto | Cantidad / Ración | Unidades |
|----------|-------------------|----------|
|----------|-------------------|----------|

*Figura 22*

El funcionament a la hora d'inicialitzar, modificar o crear serà el mateix que en tots els altres casos.

#### **- 4.7.4 - Creació de menús**

La creació de menús serà exactament igual a la dels productes de venta, pel que fa a tota la interfície i funcionament. L'únic que canviarà serà el camp 'multiproducte' que haurà de poder guardar la quantitat de plats que es desitgi, quines opcions hi ha per cada plat, quines d'elles tenen suplement i de quan és el suplement. Per tant, primer de tot, necessito saber quins camps sobre aquesta informació necessito guardar, i com guardar-los. Els plats estaran ordenats numèricament (1 primer plat, 2 segon plat, etc..), i el suplement serà en euros. En principi serien 3 dades a mostrar i guardar: el 'id' del producte de venta (Es mostrarà el nom), el número de plat en què està inclòs per escollir, i el suplement que tindria si el seleccionen (Per defecte 0). Com segueixen sent 3 dades, les guardaré amb els separadors igual que els productes de venta. I per mostrar-ho a la interfície posaré una taula de 3 columnes, una amb tots els productes de venta, una fent referència al plat en el qual es podran escollir (Per defecte 0 que voldrà dir que no estan inclosos)

i una altre per el suplement (Per defecte 0). Llavors caldrà una rutina que a cada modificació ordeni els plats (Per si de cas l'usuari posa 1er i 3er plat, però no 2on, la rutina els ordenarà de més gran a més petit i els posarà en llista començant per 1,2,3....), i generi la cadena de text a guardar a la base de dades.

Cal comentar també els camps visuals que apareixien als productes de venta, el cost suma, i el preu de venta suma. Per tal de poder mostrar informació que serveixi, es faran els càlculs seleccionant per cada plat, només el producte més costós, o amb el preu de venta més alt.

Per la quantitat, és el mateix tema que amb els productes de venta, es podria utilitzar per fer promocions limitades, però de moment no ens interessa, ni és cap mena de requeriment.

L'aspecte visual final del formulari seria com a la *figura 23*.

*Figura 23*

#### **- 4.7.5 - Explorador de magatzem**

M'interessa explicar aquest explorador, ja que serà casi el mateix que s'utilitzarà per el TPV. El tipus del control del explorador serà ListView, que afegint un control ImageList, podem enllaçar els objectes del ListView amb Imatges afegides a la col·lecció d'imatges ImageList. El tipus ListView ens permet explorar

objectes igual que si fos un explorador de Windows, seleccionant el tipus de vista que volem (Mosaic, Icones, llista o detall), i ens dona la possibilitat de generar codi en el moment de cada succés d'interacció usuari/interfície. Hauré de fer una rutina que carregui tots els productes com a objectes del ListView, carregui les imatges com a objectes del ImageList, i els enllaci. Totes les opcions estaran en botons, que interactuaran segons el producte que tinguem seleccionat. Per tant, només interactuarem amb l'explorador amb el doble clic, que per defecte modificarà el producte seleccionat, o si és carpeta canviarem la comanda SQL de consulta de productes amb la clàusula corresponent al producte pare igual a la 'id' de la carpeta a la que hem fet doble clic. Com ja tenim la vista detallada dels productes en base de dades, l'aspecte visual de l'explorador la farem més directe e intuïtiva (Tot icones).

Per últim haurem de tenir alguna imatge per defecte, per si un producte no té imatge, i més important encara, per si una carpeta no té imatge, i poder-la diferenciar. Un exemple de com quedaria l'explorador amb imatges carregades es pot veure a la *figura 24*.



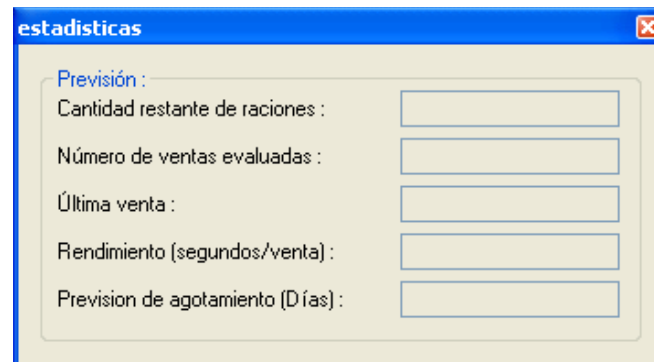
*Figura 24*

- Això es un exemple amb productes aleatoris ingressats per fer proves -

La possibilitat d'organitzar-se per carpetes és totalment elecció del usuari.

#### - 4.7.6 - Formulari detall d'estadístiques

No requereix cap explicació, serà un formulari on mostrarem les cinc dades guardades a la base de dades sobre el producte seleccionat. L'aspecte que tindrà es pot veure a la *figura 25*.

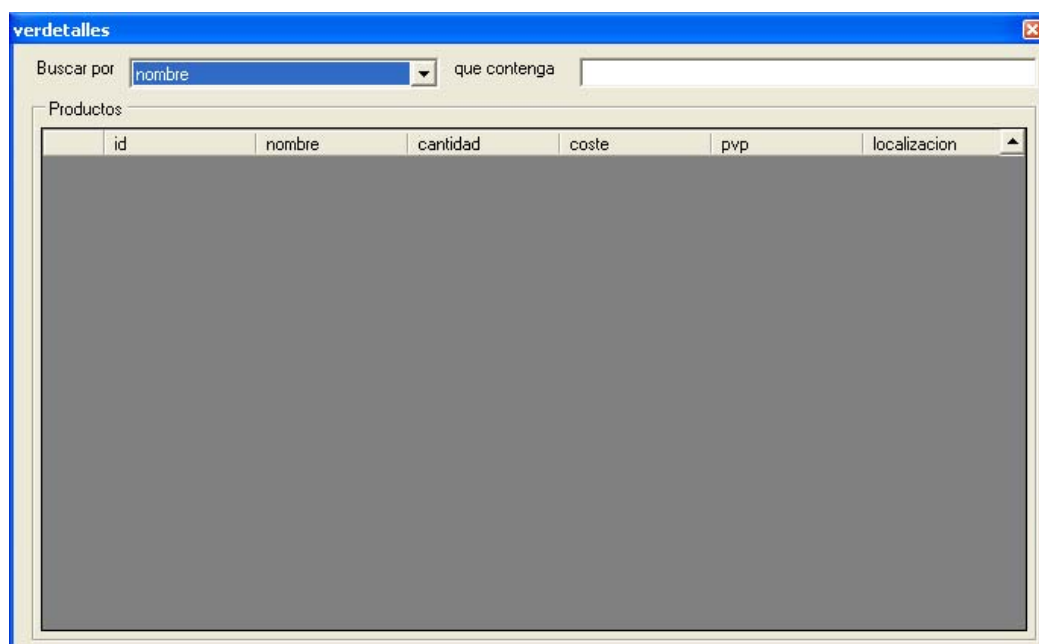


The screenshot shows a window titled 'estadísticas' with a blue title bar. Inside, there is a section labeled 'Previsión:' containing five input fields with corresponding labels: 'Cantidad restante de raciones:', 'Número de ventas evaluadas:', 'Última venta:', 'Rendimiento (segundos/venta):', and 'Prevision de agotamiento (Días):'.

*Figura 25*

#### - 4.7.7 - Vista detall

Aquest altre formulari tampoc requereix cap explicació, serà tot una taula on mostrarà tota la informació de la base de dades. L'únic que afegiré serà la possibilitat de buscar text o informació en alguns dels camps, per si necessiten trobar algun producte en concret ràpidament. L'aspecte serà el següent :



The screenshot shows a window titled 'verdetalles' with a blue title bar. It features a search bar at the top with a dropdown menu set to 'nombre' and a text input field labeled 'que contenga'. Below the search bar is a section labeled 'Productos' containing a table with the following columns: 'id', 'nombre', 'cantidad', 'coste', 'pvp', and 'localizacion'. The table body is currently empty.

*Figura 26*

En el menú llistat de dalt a l'esquerra, com veiem a la *figura 26*, introduïrem tots els camps possibles, i buscarem els productes que en aquell camp tinguin el que s'escrigui en el camp de text.

#### - 4.7.8 - Entrada de productes

Un cop tenim tots els productes entrats, necessitem poder fer entrada des de facturació de cadascun d'ells, per avaluar preus i tenir unes bones previsions i estadístiques (Podríem funcionar sense fer entrada de res, i tindríem rendiments igual).

Crearem dos formularis nous per tal de donar entrada als productes, només quan tinguem seleccionat un producte cru al explorador del magatzem deixaré poder entrar. El primer serà amb totes les opcions (*Figura 43*) de les factures d'entrada (Taula factures rebudes), juntament amb les dades del producte cru corresponent. Afegiré un quadre de selecció de proveïdors (Agafar dades de la taula corresponent) amb la opció d'obrir el segon formulari, que serà de gestió de proveïdors.

The image shows a software window titled 'entsalprod' with a close button. Inside, the main heading is 'ENTRADA Sucre'. Below this, there are several input fields and controls:

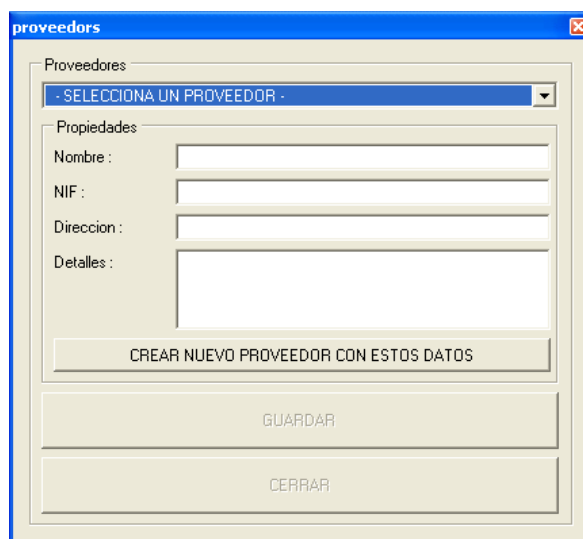
- Proveedor:** A dropdown menu with the text '- SELECCIONA UN PROVEEDOR -' and a button labeled 'M'.
- Nº Factura:** A text input field.
- Cantidad:** A text input field with a value of '0' and a small increment/decrement control.
- Coste:** A text input field with a value of '0,00' and a small increment/decrement control.
- IVA:** A text input field with a value of '0,00' and a small increment/decrement control, followed by a '%' symbol.
- Coste total:** A text input field with a value of '0,00' and a small increment/decrement control.
- Unidades:** A text input field with the value 'Gramos'.
- Coste unidad:** A text input field with a value of '0,000000000000000000' and a small increment/decrement control.
- PVP(aprox):** A text input field with a value of '0,063278700000000000' and a small increment/decrement control.
- Detalles:** A large text area for additional information.
- Aceptar:** A button at the bottom of the form.

Figura 43  
- 62 -

Faré el segon formulari (*Figura 44*) amb totes les opcions d'un proveïdor (Taula proveïdors) i faré el formulari internament igual al de control d'usuaris (Variable global per modificació/inserció i quadre de selecció), per tal de poder gestionar-los.

En el formulari d'inserció calcularé a partir del preu d'entrada total, el cost per unitat del producte, així sabrem el cost real de cada unitat del producte cru a la hora de visualitzar-ho en els demás tipus (Per els costos dels productes de venta i menús). Per tal de fer una actualització general de tots els productes, hauré de consultar tota la taula de productes de tipus venta o menú, comprovar la seva cadena de text 'multiproducte', per saber si contenen aquell producte cru, o corresponentment al menú, aquell producte de venta.

El PVP aproximat el calcularem mirant la mitja de PVP de tots els productes que continguin aquell producte cru, aquesta serà una funció que duré a terme cada vegada que es modifiqui qualsevol preu d'un producte de menú (Actualitzar productes de venta), i cada cop que es modifiqui un producte de venta (Actualitzar productes crus), totalment invers als costos.



*Figura 44*

## ► 4.8 - Formulari TPV

Aquest formulari, al igual que el del magatzem, té molts requeriments, i per tant serà molt extens. Ha de ser tota la plataforma de treball per efectuar les ventes, per tant ha de tenir explorador de productes per vendre, l'estat de la factura actual, la possibilitat de gestionar encàrrecs, gestionar clients i tot tipus de tràmits amb caixa (Agafar canvi, ingressar i cobraments). Una primera vista de la interfície que ens interessa es pot observar a la *figura 27*.

Figura 27

El formulari tindrà una mida fixada de 1024x768 per agafar tota la pantalla del monitor tàctil. He afegit l'explorador, els botons per seleccionar el client, facturar, gestionar i crear encàrrecs, agafar canvi o ingressar a caixa, facturar amb descompta i per sortir. Com només hi ha una taula per poder mostrar la informació

de la factura actual, faré que es pugui interaccionar directament amb aquesta taula per afegir o treure productes.

#### **- 4.8.1 - Format de factura / encàrrec**

Les dades a gestionar seran les mateixes per un encàrrec que per la factura final, ja que només hauré d'afegir les dades sobre el cobrament, que haurà de poder ser efectuat per parts, si es desitja. Les dades més bàsiques a guardar són: el client (Guardarem només la 'id'), la data i hora del encàrrec/factura (Comanda SQL NOW()) i l'usuari que l'ha generat.

En el cas dels productes, tenim un camp de text per guardar-ne tota la informació. Tenim dos tipus de productes que podrem vendre, els productes de venda normals i els menús. Com el preu o la informació dels productes al llarg del temps pot variar, però en una factura no, tindrà que guardar el nom i el preu actuals del producte a la informació de la factura. Així com també afegiré la quantitat que se'n ven i el preu final per agilitzar-ne lectures (També afegiré la 'id' per especular sobre estadístiques). Per tant, per guardar un producte de venda utilitzarem una cadena de text amb separadors per la informació descrita, i uns altres separadors entre productes.

Pels menús serà diferent, ja que he de guardar quin plat de cada possible elecció del menú ha agafat el client, i quan tenia de suplement. Per tant, seguiré el mateix format que fins ara, i en el camp separat del nom del producte (Dins de la cadena de text), inclouré una altra cadena de text amb separadors on anirà tota la informació de l'elecció de menú del client, i al igual que per els productes de venda, després vindrà la quantitat, preu i preu final (I la 'id' per les estadístiques).

| Nombre              | Can. | Pvp(u) | Pvp  |
|---------------------|------|--------|------|
| Buñuelos de Bacalao | 1    | 3,40   | 3,40 |
| Alitas de Pollo     | 3    | 2,75   | 8,25 |
| Ensalada 3 Colores  | 1    | 3,25   | 3,25 |
| Ensalada Alemana    | 2    | 3,25   | 6,5  |
|                     |      |        |      |

*Figura 28*

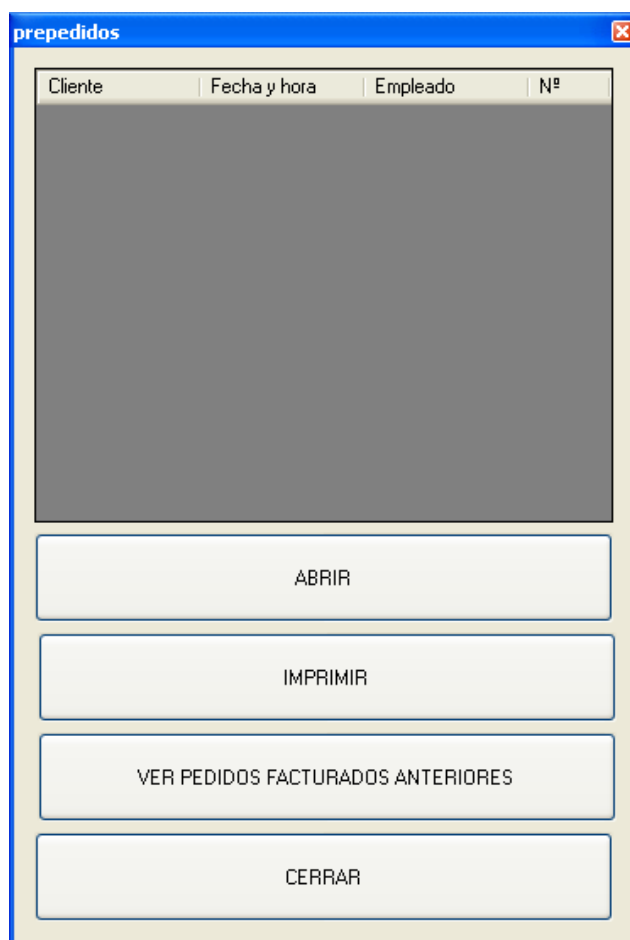
A la *figura 28* podem observar com quedaria la cadena de text carregada sobre la taula. Les línees les faig més grans per dal de poder-les pitjar amb el dit en el monitor tàctil.

#### **- 4.8.2 – Gestió d'encàrrecs**

Com tota la informació dels encàrrecs i factures abans de cobrar és la mateixa, gestionaré la factura que s'estigui tramitant com un encàrrec més, i crearé una variable global 'id' que indiqui sobre quin encàrrec s'està treballant. D'aquesta manera, simplement afegint un formulari extra per gestionar i indicar sobre quin encàrrec es vol treballar, ja tindrem el sistema d'encàrrecs fet. Per enviar com encàrrec la factura actual, el botó d'encàrrec, simplement guardarà l'encàrrec actual, generarà un encàrrec buit, i indicarà a la variable global que treballen amb l'encàrrec buit.

Com es pot observar a la *figura 29*, he afegit un botó per tal de poder observar encàrrecs ja facturats, per si es volgués veure alguna factura antiga al moment de vendre. L'apartat d'impressió el comentaré més tard juntament amb la facturació i cobro d'un encàrrec, ja que es farà servir la mateixa funció.

Si enllaço el botó de la *figura 27*, de veure encàrrecs, amb el formulari de la *figura 29*, i enllaço la 'id' del encàrrec seleccionat amb la variable global del encàrrec amb el que estem treballant, ja estarà tot fet.



*Figura 29*

#### **- 4.8.3 – Selecció d'un producte de venta o menú**

En el moment en que l'usuari del TPV prem un producte en el explorador, primer de tot s'haurà de mirar el tipus d'aquest producte.

Si és del tipus producte de venta, mirarem totes les matèries primes que utilitza, les quantitats que queden, i els càlculs de rendiment, per saber si li queda poc per acabar-se (En tals casos es mostraria avís). Un cop sabem el tipus de producte, i sabem que encara en queda, de moment procedirem només a extreure la

quantitat de matèria prima que el producte de venta ens indica a cadascuna de les que correspongui. Afegint així al encàrrec amb el que estem treballant la línea corresponent o augmentant la quantitat de la línea que correspongui si ja estava afegit.

En canvi, si es del tipus menú, necessitem una interfície per tal de que al treballador li sigui còmode demanar quins plats desitja el client, i que quedi tot ben apuntat.

The screenshot shows a software window titled "Opciones para menú" with a blue title bar. Inside the window, there is a large text input field labeled "Titulo" and a "SALIR" button. Below these, there are two table-like structures. The first is labeled "Opciones para el plato X" and has columns "Productos", "Suplemento", and "Agotado en...(Dias)". The second is labeled "Seleccionado" and has columns "Producto", "Orden", and "Suplemento". Both tables have empty rows below their headers.

Figura 30

Per tant, farem servir la interfície de la *figura 30*. En aquesta interfície començarem per el plat 1 sempre, i anirem carregant totes les opcions, un cop seleccionat el plat X, s'afegirà a la taula inferior, on seleccionant-lo podrem eliminar-lo i tornar a la selecció d'aquell plat. Un cop seleccionats tots els plats, d'interfície retornarà al TPV la cadena corresponent a tota la elecció, i es tancarà automàticament. Un cop el TPV rebi tota la informació, mirarà quins són tots els productes de venda seleccionats, i procedirà a la seva posterior selecció igual que he explicat anteriorment.

També cal tenir en compte que cada vegada que afegim un producte, sigui del tipus que sigui, es mostrarà per la pantalla del Visor VFD el seu nom, quantitat i preu. Es farà utilitzant la funció del mòdul de suport.

#### **- 4.8.4 - Canvi i ingrés a caixa**

El canvi i ingrés a caixa van totalment relacionats amb el caixer, el qual està endollat a la impressora per fer-ne una obertura automàtica. L'obertura automàtica del caixer funciona simplement enviant una cadena de caràcters per la impressora. S'ha de vigilar molt, ja que molts softwares envien les cadenes de caràcters com a text degut al format de lletra, i en aquest cas s'hauran d'enviar en ASCII cru.

Per el canvi només necessitem obrir el caixer, i que el treballador faci la seva feina, però en el cas del ingrés necessitem que el treballador pugui escollir la quantitat que ingressa a caixa per donar canvi, ja que això a de quedar relaxat a la base de dades. Aquí és on utilitzarem un formulari molt genèric que mostri la quantitat de diners a caixa per canvi (Variable fixa a la base de dades) i preguntí quan es vol ingressar. Es pot observar a la *figura 32*.

The image shows a software window titled "Ingresar dinero a caja". The window has a yellow background and a blue border. At the top, it displays "CANVIO CAJA = 0,00" and "INGRESAR =" followed by an empty text box. Below this is a numeric keypad with buttons for digits 0-9, a left arrow, and a "CI" button. To the right of the keypad are two large buttons: "CANCELAR" and "OK". At the bottom, it displays "TOTAL DESPUES = 0,00".

Figura 32

El funcionament del teclat numèric és molt bàsic, no cal fer cap explicació. Un cop seleccionat el import a ingressar, es sumarà a la base de dades i ja tindran fet d'ingrés.

#### - 4.8.5 - Selecció i gestió de clients

En el formulari TPV podem observar que es carrega la informació de la 'id' del client seleccionat. Per tal de gestionar-los i a la vegada buscar-los i seleccionar-los, hauré de crear un nou formulari per buscar i crear clients. Aquest formulari el podem observar a la figura 33, per crear un nou client només cal omplir els camps i pitjar el botó, i simplement es generarà la funció SQL que ingressarà el client a la base de dades, i passarà la 'id' del nou client al TPV. I si en canvi desitgen buscar un client ja creat, s'ha de marcar els quadres de selecció del costat dels quadres de text per els quals es vulgui buscar, i indicar què és vol buscar. Això es genera fàcilment amb una creació per nivells de la consulta SQL de consulta de clients a la base de dades, segons els quadres de selecció, per tal d'omplir la taula inferior a la figura 33.



*Figura 33*

#### **- 4.8.6 - Cobrament**

Ara que ja he explica tots els tràmits per tal de configurar un encàrrec, hem falta poder cobrar-lo per poder facturar-lo i canviar-lo de taula. Ja sigui amb descompte o sense, el procediment serà el mateix. Primer de tot ens fa falta un formulari per efectuar el cobrament, i saber com guardarem la informació del cobrament multi tipus.

Com a informació de cobrament, tenim una cadena de text, que la utilitzarem amb separadors com fins ara amb altres camps, diferenciarem el tipus de cobro per 0 efectiu, 1 targeta i 2 tiquets. A la cadena guardarem amb separadors a mida que es vagi cobrant, la forma de pago i la quantitat. A cada pagament (Parcial o total) s'han d'actualitzar les dades del caixer, i quan arribi a cobrir el preu total, en el cas de ser efectiu es tornarà el canvi (Amb tiquets no es torna canvi, i amb targeta

només es pot cobrar la totalitat). Tot actualitzant a temps real cada pas al formulari de cobrament de la *figura 34*, i esperant a que el treballador accepti.

The screenshot shows a software window titled "Cobro" with a yellow background and a blue border. At the top, a white box contains the text "INGRESAR DINERO A CAJA". Below this, the interface displays three lines of text: "CAMBIO CAJA = 0,00", "TOTAL PEDIDO = 0,00", and "RECIBIDO =" followed by an empty text input field. Underneath these fields are three radio buttons labeled "Efectivo", "Tarjeta de credito", and "Tickets". Below the radio buttons is a numeric keypad with buttons for digits 0-9, a "CI" button, and a "CANCEL" button. To the right of the keypad is a large "OK" button. At the bottom of the window, the text "CANVIO = 0,00" is displayed.

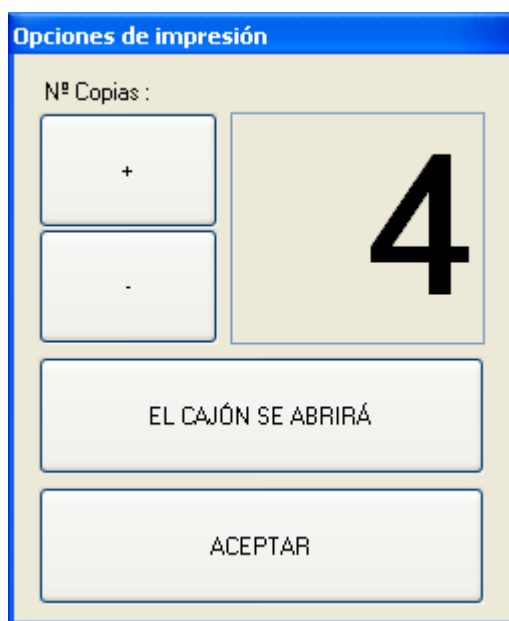
Figura 34

El formulari s'inicialitzarà amb els valors corresponents a cada quadre de text passats des del formulari TPV. Com a variables globals tindrem el total de la factura, la cadena de text del pagament i el canvi. Cada cop que alguna tecla es premi (Ja que s'actualitza el pagament), s'actualitzaran tots els valors i es modificarà el Visor VFD, per tal de mostrar detalladament el pagament en tot moment. Com es pot comprovar a la part superior de la *figura 34*, també incloc la possibilitat d'ingressar diners a caixa, que ingressant 0 seria per agafar canvi, i només tinc que cridar el formulari d'escrit anteriorment. Un cop està tot cobrat, i el canvi entregat, només cal

la instrucció SQL que ingressarà la factura amb les dades corresponents al cobro i al encàrrec a la taula de les factures.

#### **- 4.8.7 - Impressió de tiquets**

Un cop ja està cobrat i facturat, necessitem imprimir els tiquets de la factura. La quantitat de tiquets necessaris pot ser variable, per tant afegiré un formulari entremig per escollir la quantitat de còpies.

El formulari té un títol "Opciones de impresión" a la part superior. A sota, hi ha el text "Nº Copias :". A la dreta d'aquest text hi ha un botó amb el símbol "+". A la part inferior esquerra hi ha un botó amb el símbol "-". A la dreta d'aquests dos botons hi ha un rectangle gran que mostra el número "4". A sota d'aquests elements hi ha dos botons: "EL CAJÓN SE ABRIRÁ" i "ACEPTAR".

*Figura 35*

A la *figura 35* podem observar el formulari d'elecció de còpies. És molt simple, només ha de recollir de la interfície les dades, que fàcilment es poden seleccionar amb el monitor tàctil, i enviar-les a la rutina d'impressió.

La rutina d'impressió ens ve donada per la classe PrintDocument inclosa a les llibreries d'impressió System.Drawing.Printing del .NET Framework 2.0. Per tal de fer-la funcionar be, hem de tornar a declarar la seva funció PrintPage que dibuixa tota la pàgina a imprimir, incloent-l'hi totes les declaracions i mètodes que vulguem que utilitzi per dibuixar tot el tiquet. La impressió vindrà donada per la variable global

del tiquet amb que treballem, i a partir d'aquí, amb les funcions que ens ofereix el .NET Framework 2.0 per dibuixar pàgines (Per escriure text i adjuntar imatges), intentant quadrar-ho tot amb variables que serviran de guies, anirem omplint tots els camps necessaris. A la *figura 36* es pot observar com queda un tiquet després de fer moltes proves (Quadrar-ho amb el paper), i parlar-ho molt amb el client final.

**LOGOTIP**

C.Cial Diagonal Mar  
08019 - Barcelona  
Chic And Chips S.A. A-64183171

4k-pc-2-8761 14/06/2007 1:46:50

| Unid.       | Descripción       | Precio | Total |
|-------------|-------------------|--------|-------|
| 2           | Buñuelos de Ba... | 3,40   | 6,80  |
| 1           | Ensalada Alema... | 3,25   | 3,25  |
| 1           | Ensalada 3 Col... | 3,25   | 3,25  |
| 1           | Alitas de Poll... | 2,75   | 2,75  |
| 3           | Calamares Roma... | 5,50   | 16,50 |
| SUBTOTAL    |                   |        | 32,55 |
| IVA         |                   |        | 2,28  |
| TOTAL EUROS |                   |        | 34,83 |
| ENTREGADO   |                   |        | 34,83 |
| SU CAMBIO   |                   |        | 0,00  |
|             | Efectivo          | 5,00   |       |
|             | Efectivo          | 6,30   |       |
|             | Tickets           | 2,00   |       |
|             | Tarjeta           | 21,53  |       |

GRACIAS POR SU VISITA

---TURNO---

**61**

-----

*Figura 36*

Com podem observar a la *figura 36*, el tiquet inclou totes les dades de la factura, del cobrament (També si és fet per parts), de la corresponent empresa, adjunta el logotip i un últim número denominat torn. El torn pot servir per fer recollides, i ve donat exclusivament per els dos primers nombres enters del 'id' de la factura.

#### **- 4.8.8 - Càlcul de rendiment**

Un cop està cobrat, facturat i imprès, ja sabem del tot segur que el material s'haurà gastat. No cal actualitzar estocs, ja que tot el material que s'ha gastat ja estava reservat en el encàrrec que hem passat a factura, però si cal actualitzar les dades de rendiment i esgotament.

Per tal de calcular els rendiment, primer de tot hem de tenir la mida de la mostra, i el temps transcorregut des de l'últim mostreig. S'ha de tenir en compte, que aquest procés es repetirà per cada matèria prima que s'hagi utilitzat per els productes de venda corresponents (Ja siguin de menú o no), i s'actualitzaran les variables dels corresponents menús i productes de venda segons el rendiment (Ja que tampoc es poden esgotar, però ens interessa saber cada quan es venen). Per començar, en el cas que no s'hagi venut mai aquell producte, se li donarà un rendiment 0, es posarà el número de mostres a 1, i se li donarà la data de la venda corresponent. A partir de les següents mostres sempre s'aplicarà la mateixa formula, tenint en compte sempre que una sola venda d'un producte pot ser de més d'una unitat. La formula seria la de la *figura 37*.

- Càlcul del nou rendiment -

$$((\text{RendimentAntic} * \text{NúmeroMostres}) + (\text{CantitatVenuda} * \text{DiferenciaTemps}))$$

---

$$(\text{NúmeroMostres} + \text{CantitatVenuda})$$

*Figura 37*

Com es pot observar a la figura superior, el càlcul de rendiment és només un càlcul de la mitjana acumulativa sobre un nombre finit de mostres, del temps entre venda i venda d'un producte. L'aprenentatge seria molt fort quan el número de mostres es reduït, i a mida que el número de mostres augmentes, seria més difícil de

variar inclús amb canvis bruscs. I com més temps passi el càlcul seria més precís, donant a veure sempre un rendiment promig des de la creació del producte i la seva primera venda.

## ► 4.9 - Formulari de quadre de caixa

Aquest formulari ha de permetre entrar totes les dades corresponents a l'estat de la caixa, i comprovar si quadren amb l'estat de la base de dades. Per tant primer de tot, per la interfície necessitarem un teclat numèric genèric per entrar números amb el monitor tàctil. Com ja tenim el formulari per ingressar diners a caixa, el podré utilitzar exactament igual, però inicialitzat amb altres variables, per tal de demanar-li al treballador, quans diners hi ha a caixa de cada cosa.

**Caja**

Selección i control de cajas

Caja asignada a este PC: Label18 **ÚLTIMO CIERRE DE CAJA :** 00/00/0000

Actualmente en caja en efectivo :

- Pulsa en el boton grande donde pone el tipo de moneda para añadir 1 de ese tipo
- Pulsa en el boton de la derecha del tipo de moneda para quitar 1 de ese tipo

|       |   |   |       |   |   |
|-------|---|---|-------|---|---|
| +0,01 | - | 0 | +0,02 | - | 0 |
| +0,05 | - | 0 | +0,10 | - | 0 |
| +0,20 | - | 0 | +0,50 | - | 0 |
| +1    | - | 0 | +2    | - | 0 |
| +5    | - | 0 | +10   | - | 0 |
| +20   | - | 0 | +50   | - | 0 |
| +100  | - | 0 | +200  | - | 0 |
| +500  | - | 0 |       |   |   |

EFFECTIVO = 0,00

**REALIZADO POR :** Desconocido

Actualmente en caja con tarjeta o tickets :

TARJETAS = 0,00

TICKETS = 0,00

1ero Introduzca en el panel de la izquierda, todas las monedas de cada uno de los tipos que hay en la caja.

2o Introduzca el total de tarjeta de credito en caja pulsando <- al sitio correspondiente, o introduzca los cobros de tarjeta uno a uno pulsando + al sitio correspondiente.

3o Igual que antes pero con los botones de al lado de 'tickets='.

4o Pulsa CUADRAR y en el caso de que cuadre el total cobrado en efectivo, tarjeta y tickets se corresponda con el total cobrado en tickets desde el ultimo cuadro, y esto se corresponda con el total ingresado a caja en cada uno de los tipos, el cuadro será correcto.

**CUADRAR, SALIR Y GUARDAR**

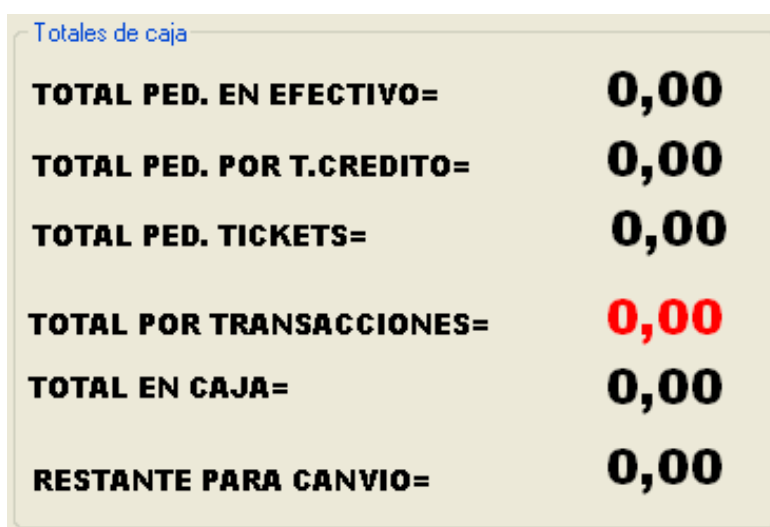
**VER FACTURACIÓN**

Figura 38

Com es pot observar a la figura 38, he influït moltes explicacions. El número d'efectiu en caixa s'introduirà moneda a moneda, i el total de tiquets o targetes s'introduirà directament el tota amb el teclat numèric anterior, o de mica en mica pitjant el botó indicat com a (+). Un cop tot introduït, al prémer el botó de quadrar i sortir és quan farà totes les comprovacions. Ha de quadrar el total d'efectiu introduït

en el quadre amb el total a caixa (Incloent el canvi i tot lo cobrat), el total de tiquets amb el total de tiquets del caixer i el total en targeta amb el total de targeta del caixer. Un cop tot quadrat, posarà tot el caixer a 0, i deixarà la mateixa quantitat per canvi que hi havia anteriorment, sobretot indicant també al treballador amb un MsgBox quina és la quantitat que ha de deixar a caixa. En el cas de que la caixa no quadri, i l'usuari sigui de màxim nivell, s'utilitzarà el formulari de la *figura 34* per preguntar quan es vol deixar a caixa, i es quadrarà tot a 0 igualment.

El botó de veure facturació només deixaré que es visualitzi en el cas de que l'usuari sigui de més de nivell 9, ja que permetrà mostrar tota la informació corresponent a com hauria d'estar la caixa (Com a la *figura 39*), i permet un tancament instantani.



|                                  |             |
|----------------------------------|-------------|
| Totales de caja                  |             |
| <b>TOTAL PED. EN EFECTIVO=</b>   | <b>0,00</b> |
| <b>TOTAL PED. POR T.CREDITO=</b> | <b>0,00</b> |
| <b>TOTAL PED. TICKETS=</b>       | <b>0,00</b> |
| <b>TOTAL POR TRANSACCIONES=</b>  | <b>0,00</b> |
| <b>TOTAL EN CAJA=</b>            | <b>0,00</b> |
| <b>RESTANTE PARA CANVIO=</b>     | <b>0,00</b> |

*Figura 39*

## ► **4.10 - Formulari de gestió**

En el formulari de gestió, el que ens interessa principalment és poder generar una vista detall en un format extern de la facturació d'entrada o de sortida d'un determinat període de temps.

Gestión: Compras / Ventas

Gestión:

**INICIO**

junio de 2007

lun mar mié jue vie sáb dom

28 29 30 31 1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 1

2 3 4 5 6 7 8

Hoy: 14/06/2007

**HASTA**

(INCLUSIVE)

Seleccionar

Deseleccionar

**FIN**

junio de 2007

lun mar mié jue vie sáb dom

28 29 30 31 1 2 3

4 5 6 7 8 9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 1

2 3 4 5 6 7 8

Hoy: 14/06/2007

Excel

SACAR REFERENCIAS DE COMPRAS A EXCEL

SACAR PEDIDOS DE VENTA A EXCEL

SACAR FACTURACION COMPLETA A EXCEL

Pedidos / Facturación

| N° Factura | Caja | Usuario | Cliente | Fecha / Hora | Forma de pago | Total | Venta Interna |
|------------|------|---------|---------|--------------|---------------|-------|---------------|
|------------|------|---------|---------|--------------|---------------|-------|---------------|

Total: Label10 Total efectivo: Label10 Total tickets: Label10

Total + IVA: Label10 Total tarjeta: Label10

Imprimir

Ver caja

Ver cliente

Ver usuario

| Producto | Cantidad | Precio (U) | Precio | ID |
|----------|----------|------------|--------|----|
|----------|----------|------------|--------|----|

Figura 40

Com es pot veure a la figura 40, el període de temps es selecciona amb dos calendaris (Inici i fi) que delimitaran la consulta SQL sobre la taula de facturació. Després amb les llibreries de vinculació entre .NET Framework 2.0 i Microsoft Excel anomenades Microsoft.Office.Interop.Excel, podré obrir un nou llibre d'excel i introduir-hi les dades corresponent a la consulta (Tota la facturació d'entrada o de sortida).

Com es pot observar a la figura 40, hi ha molts altres apartats, taules, i botons, que no tenen cap referència amb cap requeriment. Com ja vaig comentar molt al principi, aquesta aplicació està en continu desenvolupament, i actualment estic creant un sistema per buscar per tot el històric de facturació (Buscar detalls

concrets o fer estadístiques). Però com no forma part dels requeriments i està a mig fer, no ho explicaré.

## 5 – Funcionament bàsic

Per tal d'agilitzar les proves del software, i tenir una idea més clara de tot el que es pot dut a terme un cop l'aplicació ja està acabada, he decidit adjuntar un petit manual d'usuari esquematitzant com es fa cada tasca.

Per tal de poder executar el software es necessita tenir instal·lat el .NET Framework runtime 2.0, que s'instal·la automàticament amb les actualitzacions de Windows. Si no hi ha possibilitat d'actualitzacions automàtiques, el paquet ocupa 21Mb i es pot descarregar a la pàgina web de Microsoft [5].

Un cop tenim el .NET 2.0 instal·lat, només cal l'executable de l'aplicació i una base de dades MySQL on connectar. Si no es disposa d'un servidor [6] i d'un usuari amb permisos suficients, amb el software es proporciona l'estructura de dades de la base de dades en format de dades, per tal de copiar-les a un servidor MySQL recent instal·lat o en funcionament ( Ha d'estar aturat per copiar les dades).

Un cop tenim tot preparat, al arrancar l'aplicació ens preguntarà totes les dades de configuració del servidor MySQL. Quan ja estigui ben configurat podrem començar a fer anar el software.

En el cas de disposat de tots els dispositius externs, endollarem el teclat al COM1 i el Visor VFD al COM2, la impressora on vulguem.

Per entrar suposarem l'usuari SYSTEM contrasenya SYSTEM per tenir tots els privilegis. Un cop hem ingressat usant el formulari d'ingrés manual, tenim la opció de crear targeta en el formulari de control d'usuaris (Entrar al formulari, introduir la targeta, pitjar gravar targeta, i introduir el PIN de la targeta nova), en el cas de que tinguem el perifèric, i o creiem convenient.

Un cop a dins, primer de tot ens interessa crear productes, per tal de provar el funcionament, per tant anirem al magatzem. Un cop al magatzem, tenim la opció de crear carpetes per organitzar el magatzem, sinó, fàcilment amb cada botó podrem crear un producte del tipus que vulguem. Si volem un anàlisis correcte, hem de començar des de les bases, per tant començarem creant alguns productes crus (Matèries primes), per indicar què gastaran els productes que posteriorment venguem. Per tant, un cop estiguem al formulari del magatzem anirem al formulari d'ingressar producte i omplirem tots els quadres de text amb les dades corresponents. Un cop efectuat el mateix procés per tots els productes crus (Incloent imatges si s'escau), es continuarà per la creació d'alguns productes de venta (Ingressar botons), indicant les quantitats que gasta de cada producte cru a la taula corresponent, com podem observar a la *figura 41*.

| Producto              | Cantidad / Ración | Unidades |
|-----------------------|-------------------|----------|
| 116: Patatas Bravas   | 0                 |          |
| 118: Calamares Ro...  | 0                 |          |
| 119: Croquetas        | 0                 | 5        |
| 120: Alitas de Pollo  | 0                 |          |
| 121: Sandwich Ve...   | 0                 | sandwich |
| 123: Carpaccio de ... | 0                 |          |
| 125: Coca Cola Lig... | 100               | ml       |
| 126: Agua Osmosis     | 400               | ml       |
| 127: Nestea           | 0                 | ml       |
| 128: Fanta Naranja    | 0                 |          |
| 129: Fanta Limon      | 0                 | ml       |
| 131: Kaliber Sin      | 0                 | ml       |

*Figura 41*

Si volem provar el menú haurem de crear tots els productes de venta necessaris per omplir les eleccions per menú que desitgem (Segons el número de plats). Després d'inserir uns quants productes més i uns quants productes de venta

més enllaçats, procedirem a crear un menú i seleccionar quins volem que siguin cadascun dels plats que podran escollir. Com podem observar a la *figura 42*.

| Productos             | Orden | Suplemento |
|-----------------------|-------|------------|
| 5:C.Cola 500 ml Pr... | 1     | 0          |
| 6:C.Cola Light 500... | 1     | 0          |
| 8:Heineken 330ml ...  | 1     | 1          |
| 10:Fanta Naranja ...  | 2     | 0          |
| 11:Fanta Limon 50...  | 2     | 0          |
| 14:Zero Lata          | 2     | 0          |
| 15:Ligh Lata          | 2     | 0          |
| 16:Nestea Limon 5...  | 3     | 0          |
| 17:Heineken Lata      | 3     | 1          |
| 19:Cafe               | 3     | 0,5        |
| 20:Cortado            | 0     | 0          |
| 21:Cafe con Hielo     | 0     | 0          |
| 23:Infusiones         | 0     | 0          |

*Figura 42*

Un cop tenim creats els productes, si desitgem bones estadístiques i prediccions, necessitarem donar entrada als productes crus, que marcaran els costos de tots els tipus de producte. Si no existeix facturació, es poden fer probes del software ingressant factures falses, com podem comprovar a la *figura 45*.

*Figura 45*

Quan ja tinguem l'entrada de tots els productes crus, podrem observar modificant els altres tipus de productes, que s'han actualitzat els costos. Si ara anem a marcar preus, i introduïm tots els PVP que volem per cada producte de venda o menú (Incloent suplement sempre farà preu mig) i veurem com s'actualitzen successivament.

Com ja tenim alguns productes introduïts amb preus i costos, anirem a intentar fer alguna factura o algun encàrrec. Entrant al TPV ja podrem veure els nostres productes de venda i menús, aquesta part ja és més intuïtiva, seleccionar el que es vol vendre, crear o seleccionar client, si vols guardar com encàrrec simplement donar-li al botó 'encàrrec' i es guardarà per poder-lo buscar més tard a 'buscar encàrrec', i podem anar provant d'entrar o treure coses, i en un altre ordinador amb el mateix software i servidor, es podria observar com baixen els estocs. Totes aquestes opcions es poden observar fàcilment a la interfície del apartat del TPV. I per mirar els estocs a l'altre ordinador hauria d'anar al magatzem i entrar a estadístiques o a modificar producte.

Un cop vulguem cobrar i acabar la venda, simplement li donarem al botó de facturar, i s'obrirà el formulari de cobrament. Com la caixa ve donada per el nom del host, si es el primer cop que arranqueu el software no tindreu res a caixa, per tant no es pot cobrar perquè no hi ha canvi, i s'haurà d'ingressar una quantitat suficient pitjant el botó d'ingressar a caixa, i utilitzant el formulari numèric d'euros.

En el formulari de cobrament podrem fer varis cobraments diferents amb efectiu o tiquets (Marcar quantitat i OK), o cobrar el total amb targeta (Marcar targeta i OK). Després, sortirà el formulari d'impressió, i si tenim els perifèrics corresponents o compatibles, un cop acceptem, veurem com s'obre el calaix (Si es selecciona), i surten les còpies desitjades de factures.

Aquí podríem anar provant de fer varies entrades i sortides de facturació (Ventes i entrades de productes), i més tard es podria observar com les estadístiques s'actualitzen, i es va calculant el rendiment, i genera els avisos corresponents.

Un cop ja haguem comprat o venut bastant (El negoci va funcionant), a final de dia convé quadrar caixa. Si entrem al formulari corresponent, podrem observar com les dades de tot el que s'ha venut i en el tipus de pagament en que s'han venut, concorden amb les dades guardades a caixa.

Tot aquest procés seria com un dia o torn de feina efectuat ràpid. A partir d'aquí es poden provar les opcions que s'ofereixen al propietari del negoci amb un tercer ordinador de control, per extreure la facturació en el formulari de gestió, i observar moviments de qualsevol tipus a temps real, de tots els encàrrecs efectuats, ventes, despeses, i guanys que tingui el negoci utilitzant també el formulari de gestionar encàrrecs, i el de magatzem.

## 6 - Conclusions

Com ja comentava a la introducció, aquest software m'ha aportat molt per aprendre a trobar solucions informàtiques, per persones no informàtiques, en un entorn de treball complex (Informatitzar un negoci des de zero). Realment considero que aquesta tasca, juntament amb el corresponent desenvolupament d'esquemes de disseny i funcionament de cadascun dels apartats (No només visuals, ja que cada funció podria tenir el seu esquema i podrien ser centenars, i tots han d'estar enllaçats el millor possible), es realment la tasca d'un enginyer informàtic (Es podria adaptar a casi tots els àmbits la mateixa forma de treball).

Crec que els requeriments han estat tots assolits, ja que aquest software dona garanties d'una metodologia de treball al negoci correcte, i una seguretat notable. Clarament es podria ampliar per molts racons, principalment per visualització d'històrics de tot tipus, però tampoc m'introduiria res nou, serien més formularis d'informació treballant contra una base de dades.

Afegir també que després d'acabar tot el projecte, tota la interfície hem sembla molt poca cosa per tot el codi que hi ha generat darrera. Vaig estar pensant en si podria haver reduït codi amb una millor organització, però realment, tenir en compte tots els botons i opcions que pot arribar a pitjar un usuari en una interfície com aquesta, per obligació ha de tenir un codi molt extens sobre successos i funcions a les que criden.

Apart he après molt de programació i bases de dades, però realment considero, i sobretot després d'introduir-me al desenvolupament de software juntament amb un negoci, que lo important no és aprendre molt, sinó aprendre ràpid

i saber-te adaptar a les situacions, necessitats i requeriments, per tenir una millor visió del que es planteja, i crear la millor solució possible al problema (Sense dependre dels coneixements anteriors, saber veure un esquema de funcionament i plasmar-lo en el sistema que vulgui o requereixi).

## Referències

[1]

[http://solutions.3m.com/wps/portal/3M/en\\_US/3MTouchSystems/TS/?WT.mc\\_id=www.3m.com/3mtouchsystems](http://solutions.3m.com/wps/portal/3M/en_US/3MTouchSystems/TS/?WT.mc_id=www.3m.com/3mtouchsystems)

[2]

<http://sd.epson.es/products/tms/t88IV/t88IV.htm>

[3]

Documentació:

<http://dev.mysql.com/doc/refman/5.0/en/connector-net.html>

Descarrega:

<http://dev.mysql.com/downloads/connector/net/5.0.html>

[4]

Documentació MSDN ( Inclou .NET Framework 2.0 ):

<http://msdn2.microsoft.com/es-es/library/default.aspx>

També caldria fer referència a la documentació inclosa en el Visual Studio que no puc adjuntar aquí, encara que molta part esta continguda a la Web.

[5]

<http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5>

[6]

<http://dev.mysql.com/downloads/mysql/5.0.html>

## RESUM

De una idea, unes paraules, un esquema, una aplicació. En aquest projecte es descriu tot el procés per tal de crear una solució integral per el funcionament diari d'un negoci d'hostaleria. Es consideren per igual les tasques de saber com ha de ser el funcionament, i com plasmar-lo a l'aplicació. Es podrà observar com des d'uns certs requeriments funcionals i no funcionals, genero el software amb els enllaços a tots els perifèrics requerits: Impressora de tiquets, Visor VFD, targetes chip, caixer moneder i monitor tàctil. Explicaré quines tasques requereix el negoci, i com he creat cadascun dels formularis que requereixen aquestes tasques per portar el negoci al dia.

## RESUMEN

De una idea, unas palabras, un esquema, una aplicación. En este proyecto se describe todo el proceso de crear una solución integral para el funcionamiento diario de un negocio de hostelería. Se consideran por igual las tareas de saber como tiene que ser el funcionamiento, y saber como plasmarlo en la aplicación. Se podrá observar como desde unos ciertos requerimientos funcionales y no funcionales, desarrollo el software con los enlaces a todos los periféricos requeridos: Impresora de tickets, Visor VFD, tarjetas chip, cajero monedero y monitor táctil. Explicaré cuales son las tareas que requiere el negocio, i como he creado cada uno de los formularios que requieren dichas tareas para llevar el negocio al día.

## ABSTRACT

One idea, words, a scheme, an application. In this project is described all the process to create an integral solution for the daily operation of a restoration business. The tasks of knowing how the process must be and the task of shaping it into an application are considered the same way. It will be possible to observe how from a certain functional and non-functional requirements, I develop the software with the required connections to all the peripherals: Ticket printer, VFD Viewer, chip cards, exchange box and tactile monitor. I will explain which tasks the business requires, and how I created each of these forms to accomplish a good work for the day-to-day of this business.