

Treball Final de Carrera

*Programa de gestió per la Boutique
Shahrazad.*

Khadija Mourikh

Enginyeria Tècnica d'Informàtica de gestió

Directora: Cristina Borralleras

Vic, juny de 2012

Resum de Treball Final de Carrera

Enginyeria Tècnica d'Informàtica d'Informàtica de gestió

Títol: Programa de gestió per la Boutique Shahrazad

Paraules clau: Gestió, comerç, màrqueting, fidelització, productes, clients, proveïdors.

Autor: Khadija Mourikh

Direcció: Cristina Borralleras

Data: Juny de 2012

Resum

Per tal d'iniciar el meu TFC, vaig decidir començar a desenvolupar un programa que contempla la gestió diària del comerç, així com la dels productes, clients, proveïdors, vendes i compres, d'un negoci que es tracta d'una botiga de roba, sabates, bosses i complements, per tal de convertir-ho en un sistema de gestió més complet.

Actualment, l'entorn de negoci en el petit comerç està cada cop més desplaçat degut al gran creixement de les grans superfícies per part de franquícies de cadenes molt potents. A més, els clients són cada cop més exigents i el mercat està cada vegada més saturat. És per això que voldrem incloure un valor afegit al sistema, per poder donar un nou punt de vista al negoci, de tal manera que permeti nodrir la nostra relació amb el client, de manera que la col·laboració sigui satisfactòria per ambdues parts.

I per acabar s'elabora una pàgina web per tal de fer promoció i publicitat per la botiga, ja que avui en dia és molt important existir en el món virtual.

Índex

1. Context del projecte	9
1.1. Introducció	9
1.2. Objectius	10
1.2.1. Programa de gestió	10
1.2.2. Aplicació web	11
1.3. Organització de la memòria	13
2. Planificació inicial del projecte	14
3. Descripció del projecte	17
3.1. Productes i proveïdors	17
3.1.1. Catàleg de productes	17
3.2. Clients i compradors	19
3.2.1. Fitxa de client	20
3.2.2. Compte del client	22
3.2.3. Fidelització de clients	22
3.3. Venedors	24
3.4. Vendes	25
3.5. Compres	26
3.6. Llistats i resums estadístics	26
3.7. Manual d'usuari i còpies de seguretat	27
4. Avaluació tecnològica	27
4.1. Llenguatges de programació	27
4.2. SGBD	29
4.3. Seguretat	30
5. Prototip 1: catàleg de productes	30
5.1. Requeriments	30
5.1.1. Catàleg	30
5.1.2. Categoria	31
5.1.3. Marca	31
5.1.4. Color	31
5.1.5. Talla	32

5.1.6. Iva.....	32
5.1.7. Proveïdor	33
5.1.8. Producte	33
5.2. Anàlisi	34
5.3. Disseny del sistema.....	42
5.3.1. Interfícies.....	42
5.3.2. Esdeveniments.....	47
5.4. Disseny de la base de dades.....	55
5.5. Disseny del programa.....	58
6. Prototip 2: clients	62
6.1. Requeriments	62
6.1.1. Client.....	62
6.1.2. Venedor	63
6.1.3. Obsequi	63
6.2. Anàlisi	63
6.3. Disseny del sistema.....	65
6.3.1. Interfícies.....	65
6.3.2. Esdeveniments.....	69
6.4. Disseny de la base de dades.....	71
6.5. Disseny del programa.....	73
7. Prototip 3: Vendes.....	73
7.1. Requeriments	73
7.1.1. Venda.....	73
7.1.2. Línia de venda	74
7.2. Anàlisi	74
7.3. Disseny del sistema.....	76
7.3.1. Interfícies.....	76
7.3.2. Esdeveniments.....	79
7.4. Disseny de la base de dades.....	81
7.5. Disseny del programa.....	82
8. Prototip 4: Compres.....	87
8.1. Requeriments	87
8.1.1. Compra	87

8.1.2. Línia de compra	87
8.2. Anàlisi	88
8.3. Disseny del sistema.....	89
8.3.1. Interfícies.....	89
8.3.2. Esdeveniments.....	92
8.4. Disseny de la base de dades.....	94
8.5. Disseny del programa.....	96
9. Prototip 5: Caixa	97
9.1. Requeriments	97
9.1.1. Caixa	97
9.2. Anàlisi	98
9.3. Disseny del sistema.....	98
9.3.1. Interfícies	98
9.3.2. Esdeveniment.....	99
9.4. Disseny de la base de dades.....	100
9.5. Disseny del programa.....	100
10. Prototip 6: Llistats i resums estadístics.....	101
10.1. Sobre productes.....	101
10.2. Sobre clients.....	102
10.3. Sobre proveïdors	103
10.4. Sobre vendes.....	103
10.5. Sobre compres	104
10.6. Sobre caixa	104
10.7. Rànquing de vendes	104
11. Manual d'usuari	105
12. Descripció del projecte	107
12.1. Requisits funcionals	107
12.2. Característiques de l'usuari	108
13. Anàlisi.....	108
13.1. Diagrama de cassos d'ús	109
14. Tecnologies	114
14.1. HTML.....	116
14.2. CSS.....	116

14.3. JavaScript.....	116
14.4. Ajax	117
14.5. PHP.....	117
14.6. MySQL.....	117
15. Disseny i implementació.....	118
15.1. Base de dades	118
15.2. Detalls de la implementació	123
15.3. Exemples de la implementació	124
15.3.1. Accés	124
15.3.2. Connexió	126
15.3.3. Obrir imatges	127
15.3.4. Enviament de correu electrònic.....	128
15.3.5. Mostrar colors de la talla seleccionada d'un producte	130
16. Planificació final	135
17. Conclusions i possibles millores.....	136

Part I Projecte

1. Context del projecte

1.1. Introducció

Pel treball final de carrera s'ha d'escollir un tema que motivi l'alumne, ja que comporta una dedicació i un esforç molt importants durant una bona temporada.

Això em va fer plantejar l'idea d'informatitzar un petit negoci, creant un sistema a mida per a una botiga, per cobrir les necessitats bàsiques d'aquest negoci i facilitar l'activitat botiguera, així que vaig decidir partir d'aquesta idea com a base, per desenvolupar un sistema de gestió complet.

El negoci es tracta d'una botiga de roba, sabates, bosses i accessoris, que no disposa de cap sistema informàtic, i per poder realitzar les tasques de la botiga tot es fa de forma manual.

Aquest projecte consistirà en la construcció d'un programa per fer la gestió de les dades que s'utilitzen per dur a terme les diferents activitats de la botiga, i l'elaboració d'una pàgina web per tal de fer promoció i publicitat per la botiga, ja el mercat actual de les aplicacions web no deixa de sorprendre'ns. cada vegada apareixen més noves eines que fan que la construcció i disseny d'un entorn web sigui més fàcil i ràpid de fer.

A mesura d'anar desenvolupant el projecte, es proposa la incorporació d'algunes funcionalitats que puguin aportar valor afegit al sistema, a part de facilitar les gestions rutinàries. Per això, es desenvolupen una sèrie d'eines que han de facilitar al comerciant no només la gestió, sinó també millorar la capacitat de venda.

1.2. Objectius

1.2.1. Programa de gestió

L'objectiu global és crear un sistema de gestió a mida per a un punt de venda, per tal de guardar tota la informació necessària pel negoci, i gestionar les diferents funcionalitats del sistema, la idea del nostre software és que sigui una eina informàtica concebuda per la gestió d'aquest comerç i adequada a les seves característiques i peculiaritats, diferenciant-se d'aquesta forma dels programes de gestió genèrics.

El software es desenvoluparà en entorn de Windows i es tracta d'una aplicació de gran simplicitat que permet controlar el negoci sense la necessitat de disposar d'experiència informàtica prèvia, fet que suposa un gran valor afegit.

El nostre programa permetrà gestionar de forma completa la botiga; vendes, compres, control d'estoc, caixa, estadístiques, etc.,.

A part d'això, caldrà afegir una sèrie d'utilitats per tal de facilitar el desenvolupament diari del negoci, aportant alhora més eficiència i seguretat.

En general voldrem:

- Fer vendes, compres i treure llistats de les diferents parts del sistema.
- Emmagatzemar informació dels proveïdors amb la seves dades de contacte i relacionar-los amb els productes que ens serveixen.
- Guardar les dades bàsiques dels clients i dels venedors per tal de poder fer-ne un seguiment.
- Orientació al màrqueting: fidelització de clients.

A partir d'aquí l'objectiu principal d'aquest projecte serà crear un sistema on gestionar les diferents dades de manera senzilla i poder servir més bé al clients.

1.2.2. Aplicació web

El nostre objectiu és el Desenvolupament d'una botiga virtual enfocada a fer comandes dels productes que es disposen a la Boutique Shahrazad. L'aplicació disposarà d'un catàleg de productes on es podran visualitzar les seves característiques més importants i per descomptat tindrem l'opció de poder-los encarregar.

Els productes estaran classificats per catàleg i dins de cada catàleg per les categories corresponents, a més per poder facilitar la localització dels productes de Boutique Shahrazad, s'oferirà un cercador per fer la recerca per paraules clau.

L'aplicació constarà de tres parts diferenciades degut als diferents tipus d'usuaris:

L'usuari públic serà el que té menys funcionalitats, podrà navegar per La web, veure els productes i les seves característiques, però no podrà efectuar comandes fins que no s'hagi registrat.

Un cop registrat es convertirà en un usuari registrat, i per poder encarregar productes s'haurà d'identificar i ja podrà realitzar comandes afegint o esborrant productes i especificant les unitats que desitja. A més Podrà consultar i modificar en qualsevol moment el contingut de la seva cistella de comandes i acceptar-la, també tindrà l'opció de visualitzar l'històric de les seves comandes i podrà modificar les seves dades personals.

L'Usuari administrador és el que s'encarrega de la gestió de productes, catàlegs, categories, i els diferents elements del sistema, i també tindrà l'opció de visualitzar les comandes i canviar-ne l'estat.

En resum es tracta de desenvolupar una aplicació per promocionar els productes que es disposen a la botiga a milers de nous clients de forma interactiva, de manera que els clients disposin d'una cistella de comandes per encarregar productes, amb la comoditat de fer-les des de casa seva, o des de qualsevol ordinador amb connexió a Internet, així com ajudar a la seva gestió d'una forma senzilla i clara pels usuaris i per l'administrador de l'aplicació.

Els objectius concrets consistiran en:

- Mostrar el catàleg dels Productes als possibles clients.
- Permetre encarregar els diferents productes de l'aplicació.
- Facilitar el manteniment d'aquest catàleg.

1.3. Organització de la memòria

La memòria està organitzada en quatre parts i dins de cada part en punts dels quals detallarem els més importants:

- La primera part consisteix en una explicació general de què consistirà el projecte informàtic, i la periodicitat requerida per desenvolupar-lo, i s'explica segons aquest ordre:
 1. Context del projecte: En aquest capítol s'introdueix el projecte exposant els seus objectius.
 2. Planificació del projecte: Calendari previst pel desenvolupament del projecte segons el mètode de treball escollit pel seu desenvolupar.
- La segona part consisteix en fer l'especificació, l'anàlisi i el disseny detallat del programa de gestió, i hi trobem:
 3. Descripció del projecte: Aquí hi trobem el detall de tots els conceptes desenvolupats en el projecte.
 4. Avaluació tecnològica: Resum de l'estudi realitzat a diferents llenguatges de programació, sistemes gestors de bases de dades,

i l'elecció d'aquests per a la implementació de l'aplicació, i explicació de les mesures de seguretat.

5. Prototip I: Especificació i disseny del prototip dedicat al catàleg de productes i la gestió de proveïdors, catàlegs, categories, marques, talles, colors i tipus d'Iva.
6. Prototip II: Especificació i disseny del prototip dedicat a la gestió de clients i els seus obsequis i també els venedors.
7. Prototip III: Especificació i disseny del prototip sobre vendes i facturació.
8. Prototip IV: Especificació i disseny del prototip sobre les compres.
9. Prototip V: Especificació i disseny del prototip dedicat a fer caixa.
10. Prototip VI: Especificació i disseny del prototip que conté les llistats i els resums estadístics.

- La tercera part consisteix en fer l'especificació, l'anàlisi i el disseny detallat de l'aplicació web, i hi trobem:

11. Descripció del projecte: Aquí hi trobem el detall de tots els conceptes desenvolupats en el projecte, els requisits funcionals i les característiques dels usuaris.
12. Anàlisi del projecte: Anàlisi detallat de l'aplicació.
13. Tecnologies: Detall de les tecnologies utilitzades per la implementació de l'aplicació web.
14. Disseny i implementació: Disseny i implementació de l'aplicació web.

- La quarta part consisteix en les conclusions i treball futur:

15. Planificació final: Descripció del calendari real un cop finalitzat el projecte i el contrast respecte la versió inicial.
16. Conclusions i possibles millores: Les valoracions personals del projecte. A més, una sèrie de propostes de millora i complementació del sistema.

2. Planificació inicial del projecte

El mètode de treball seguit per desenvolupar el projecte està basat en el procés de desenvolupament de les diferents parts del sistema dividides en prototipus. Aquesta opció consisteix en desenvolupar tot el sistema informàtic, en una primera versió, que és molt generalista. Llavors es posa a disposició dels usuaris perquè l'avaluïn i en diguin les mancances o les parts errònies. S'agafa tota la informació obtinguda i es torna a desenvolupar el sistema tenint-la en compte. Es presenta una altra vegada a l'usuari i es torna a repetir l'anterior fins a arribar a tenir el sistema desitjat per part de l'usuari.

Nosaltres per fer l'aplicació dividirem el sistema en diferents parts, i per cada part seguirem el model esmentat anteriorment. Aquest aporta dues grans virtuts, una de cara al programador i l'altra a l'evolució del sistema en si. La primera és la capacitat de dinamitzar tot el procés fent-ho més fàcil d'avançar. Per l'altra banda, aquest sistema permet anar veient resultats a mesura que finalitzen els prototips, i que pot avaluar el client a mesura que s'acaben, prenent-los com a experiència de cara als següents.

El desembre de 2010 es decideix tirar endavant amb aquest projecte. El primer pas és concertar una entrevista amb el client, i a partir d'aquesta es comença a realitzar el projecte amb una planificació inicial de gairebé 10 mesos de durada, com es pot veure en la figura 3.1.

		2010																2011																							
Activitat		Des				Gen				Feb				Mar				Abr				Mai				Jun				Jul				Ago				Set			
Unitat temps: setmana		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Prototip 1	Definició i estudi	P	P	P	P	P	P																																		
	Creació base dades						P																																		
	Anàlisi							P	P	P																															
	Disseny										P	P	P																												
Prototip 2	Proves												P	P	P																										
	Anàlisi															P	P																								
	Disseny																	P	P																						
Prototip 3	Proves																			P																					
	Anàlisi																				P																				
	Disseny																					P																			
Prototip 4	Proves																					P																			
	Anàlisi																						P	P																	
	Disseny																							P	P																
Prototip 5	Proves																										P														
	Anàlisi																										P	P													
	Disseny																												P												
Prototip 6	Proves																																		P						
	Anàlisi																																								
	Disseny																																								
	Proves																																								
Pàgina web																																									
Conclusions																																					P				

P: significa el temps previst en realitzar una activitat .

Diagrama de Gantt per la planificació inicial

Part II
**Especificació, anàlisi i disseny del
programa de gestió**

3. Descripció del projecte

3.1. Productes i proveïdors

Pel que fa als productes, bàsicament volem saber-ne la descripció, el catàleg en el que estigui classificat, la categoria a la que pertany, la marca, les diferents talles i colors i el tipus d'iva, així com el preu de venda actual (no volem mantenir cap història del producte). A més, volem saber la llista de proveïdors que ens ha subministrat cada producte (un mateix producte el podem comprar a llocs diferents).

El catàleg de productes disposat actualment es pot dividir en:

- Roba.
- Sabates.
- Bosses i carteres
- Complements.

Amb l'opció d'afegir nous catàlegs i categories en qualsevol moment.

3.1.1. Catàleg de productes

Per tal de gestionar les vendes i/o les compres amb més comoditat, necessitarem un llistat amb tots els productes que tenim amb la foto corresponent en cas que en desponguin. A partir d'aquest serà molt més àgil crear les vendes i/o compres i ens suposarà un estalvi de temps administratiu.

Aquest llista tindrà forma de catàleg, on hi trobarem l'informació de cada producte. La funció bàsica d'aquest catàleg és seleccionar els productes quan creem una venda.

Actualitzar el preu dels productes, serà una tasca que habitualment es farà cada vegada que es fa l'ingrés d'una compra, i al recepcionar els productes s'actualitzaran els preus i l'estoc de cada producte. També

hem de tenir l'opció de canviar el preu del producte des de la pròpia pantalla de venda en casos particulars.

Alta i baixa

Afegir un nou producte serà una tasca molt freqüent degut als canvis que presenta el mercat, constantment apareixen nous productes, i per anar bé s'ha de poder adaptar a les últimes novetats del mercat.

També interessa aplicar descomptes en casos d'oferta a alguns o a tots els productes, per aquest motiu haurem de tenir una opció d'assignar ofertes i descomptes als productes corresponents.

Al igual que donar d'alta un nou producte, hauríem de tenir l'opció de donar de baixa productes que no hi ha en estoc i que no volem repetir un altra cop, tant perquè no es venen com perquè no interessen.

Consulta i modificació

És molt habitual la necessitat de modificar el catàleg de productes, així com per descomptat consultar-lo. Hem de poder canviar-ne el preu així com rarament la descripció. Caldrà tenir en compte, però, que aquestes modificacions no afectaran a l'històric de vendes o factures, és a dir, que no pot canviar el nom d'un producte en una factura d'un any abans perquè ara en canviï el nom.

Ordena i busca

A l'hora de visualitzar el llistat de productes hem de tenir l'opció d'ordenar-los segons un criteri específic, que pot ser el codi, o la descripció, aquesta opció ens pot interessar alhora de buscar un producte. O també ens pot interessar buscar un producte directament, o sigui al introduir un valor i un camp segons el qual es vol fer la cerca ens hauria de sortir el producte que busquem si és que aquest producte existeix a la base de dades.

Exporta

Aquesta opció ens permet exportar el contingut del llistat de productes a Excel, això pot ser interessant alhora de voler utilitzar el llistat per fer càlculs o pel simple fet de voler tenir l'inventari de productes guardat a l'Excel .

Filtra

Aquesta opció ens permet filtrar el contingut del llistat de productes segons el catàleg/s, la/es categoria/es, la/es marca/es, el/s proveïdors, o també per productes que estan en estoc/sense i productes en oferta/tots.

Proveïdors

Inicialment la llista de proveïdors del producte estarà buida, i a mesura que anem ingressant compres on hi hagi aquest producte, s'actualitzarà la llista de proveïdors del producte, a part d'això hem de disposar a la mateixa pantalla de productes d'un botó de proveïdors del producte per tal de poder afegir un nou proveïdor al producte en cas necessari, per fer-ho se'ns hauria de mostrar una llista dels proveïdors dels que disposem per poder seleccionar algun d'ells i en cas que no hi sigui hauríem de tenir l'opció d'afegir-lo, també hem de tenir l'opció d'eliminar un proveïdor de la llista de proveïdors del producte, això passa per exemple en cas que el proveïdor deixa de servir un determinat tipus de productes.

3.2. Clients i compradors

Cal tenir en compte que no podem forçar a totes les persones que compren a l'establiment a cedir les seves dades i per tant, a estudiar-ne el comportament; tan pel fet de ser persones que compren puntualment, com pel simple fet de no desitjar formar part de la base de dades del negoci. Així que haurem de diferenciar entre dos perfils de consumidor: el client i el comprador.

Qualsevol persona que compra en l'establiment, ja sigui puntualment com de manera habitual, seran compradors. Per a ser clients hauran d'omplir un petit formulari amb les seves dades personals bàsiques, i amb això adquiriran un número de client que els identificarà com a tal. Aquest serà el públic que podrem estudiar.

El fet de ser client implicarà una sèrie d'avantatges, com per exemple:

- Possibilitat d'obrir un compte de client.
- Entrar en promocions i/o descomptes especials.
- Punts de client.

3.2.1. Fitxa de client

Dels clients volem guardar les següents dades personals: nom complet, DNI, adreça, població, telèfons de contacte i el correu electrònic en el cas que en disposin. A més, voldrem consultar els punts acumulats.

De cara al propietari del negoci és molt interessant tenir clients; per un costat tenir dades de contacte, i per altra banda per conèixer els hàbits de consum de la nostra clientela, i a més poder controlar les vendes, deutes i encàrrecs d'aquest. Per altra banda, el client necessitarà un al·licient per a donar les dades, així que n'obtindrà beneficis com ara entrar en promocions especials, descomptes i/o obsequis puntuals. Així com la possibilitat de tenir un compte de client.

Des de la fitxa de client haurem de poder consultar i modificar qualsevol aspecte relacionat amb els clients, de manera que hi hem de disposar de:

- Dades personals (consulta, modificació i eliminació).
- Dades comercials.
- Accés al compte de client.
- Els punts de client disponibles, totals i l'històric de regals i/o descomptes obtinguts amb la data pertinent.

- Control del regal de fidelització.
- Historial d'obsequis del client.

Alta i baixa

Per poder tenir tota aquesta informació, necessitem la funcionalitat d'afegir clients quan sigui convenient. Afegirem un nou client quan aquest hagi manifestat la voluntat d'esdevenir-ho, ja sigui en una promoció com en qualsevol altre moment. Per a fer-ho, haurem de demanar-li que ompli un petit formulari amb les dades personals que ja hem detallat, on hi consti el text legal que l'informarà de la llei de protecció de dades que serà necessari que hagi donat el consentiment.

Un cop introduïdes les dades, es posaran a zero els comptadors de punts de clients i de deutes. Donar de baixa clients no serà fonamental, però hem de tenir l'opció de fer-ho, per casos particulars.

Consulta i modificació

Dels clients ens pot interessar consultar la fitxa amb tota la seva informació, tant les dades personals, els punts de client, el seu compte en cas que en tingui. Les dades personals seran modificables i substituiran les actuals. Les dades comercials s'actualitzaran automàticament. Des d'aquí també hem de poder consultar l'estat de la resta de funcionalitats i modificar-les. Per exemple, en el cas d'haver de liquidar un compte manualment ho hem de poder fer des de la mateixa fitxa de client.

Ordena i busca

Alhora de visualitzar el llistat de clients al igual que els productes, hem de tenir l'opció d'ordenar els clients segons un criteri específic, que pot ser el codi, el DNI o el nom, aquesta opció ens pot interessar alhora de buscar un client. O també ens pot interessar buscar un client

directament, o sigui al introduir un valor i un camp segons el qual es vol fer la cerca ens hauria de sortir el client que busquem si és que aquest client existeix a la base de dades.

3.2.2. Compte del client

Hi ha un cert grup de clients habituals a qui ens interessa donar l'opció de tenir un compte per anar acumulant totes les compres que efectuen, doncs els interessa pagar-les cada determinat temps. Caldrà ser client per obrir un compte.

En el compte s'aniran acumulant les successives compres, fins a liquidar-lo, que podrà ser sota demanda o cada cert període de temps. Es a dir el propi client és qui demana en qualsevol moment de pagar els deutes, o podem fer una liquidació periòdica fixa. De totes maneres, també es pot liquidar una sola venda individualment des de la consulta d'aquesta mateixa.

A l'hora de liquidar un compte, el que farà el sistema serà generar una llistat amb el resum de totes les compres realitzades des de la més antiga de les dates que hi hagi registrada a les vendes. Quan les cobrem, el compte es buidarà automàticament.

3.2.3. Fidelització de clients

Avui en dia, l'entorn de negoci en el petit comerç es veu cada vegada més desplaçat pel vertiginós creixement de les grans superfícies o l'absorció per part de franquícies de cadenes molt potents. A més, els clients són cada vegada més exigents i el mercat està més i més saturat. És per això que voldrem dotar d'un valor afegit al sistema, i incloure el desenvolupament d'un apartat, dedicat a la fidelització de clients. Aquest ens donarà un nou punt de vista al negoci, i permetrà nodrir la nostra relació amb el client amb una gestió estratègica i dinàmica, de manera que la col·laboració sigui satisfactòria per ambdues parts. Així doncs, podem parlar d'introduir tècniques de Màrqueting per aportar valors com

compromís i confiança, ja que això ens dirigirà a obrir nous horitzons temporals a llarg termini, enlloc de centrar-nos únicament en vendre, vendre i vendre.

En definitiva, voldrem aprofitar la nostra base de dades per tal d'extreure'n les variables que intervenen en la relació amb el client i basar-nos en aquesta per tal tractar-los de manera diferenciada, enlloc de col·locar productes indiscriminadament a qui entri per la porta. Així que ens centrarem en Punts de client.

Punts de client

Els punts de client es calcularan en proporció als ingressos que aquests representin en el negoci. Inicialment no tindran punts, i a mesura que realitzin compres, aquestes es convertiran en punts a raó d'un punt per euro gastat. Aquests punts sempre seran nombres enters i si hi ha decimals en l' import total de la compra, s'arrodoniran a la baixa i es sumaran així als punts actuals i totals, que s'actualitzaran a cada venda.

La principal avantatge d'aquests punts és que serviran com a indicador per tal que els venedors tinguin una referència real de les despeses del client quan el té al davant, i a més per poder oferir quan sigui convenient descomptes o obsequis.

Per tal de controlar-los de manera automàtica, existiran unes alarmes amb avisos quan el client acumuli quantitats estandarditzades de punts que no hem bescanviat.

Això ha de quedar reflectit en algun lloc, de manera que per a cada obsequi es descomptarà la quantitat de punts que es cregui convenient dels que tingui acumulats, però sense deixar de portar el compte dels punts totals. Així doncs, a la fitxa cal que aparegui aquesta dada, tant el total de punts acumulat, com els punts actuals, que són la diferència dels totals amb els bescanviats i que s'aniran actualitzant en cada compra.

A l'hora de realitzar una acció comercial, un cop bescanviem els punts per un obsequi o descompte, s'haurà d'actualitzar també aquesta xifra i a

més, quan ho fem, se'ns haurà d'obrir una llista amb una breu descripció dels obsequis/descomptes aplicats (en cas que ja n'hi hagi), amb la quantitat de punts que s'han bescanviat i la data en la que s'ha efectuat. Això ens servirà per no repetir el mateix obsequi i poder donar més varietat a les ofertes.

3.3. Venedors

Cal guardar l'informació pertinent als venedors de l'empresa per tenir una idea sobre el rendiment de cada venedor, d'aquests ens interessa guardar el DNI, el nom, l'adreça, telèfons de contacte i el correu electrònic en el cas que en disposin.

Alta i baixa

Per poder tenir tota aquesta informació, necessitem la funcionalitat d'afegir venedors cada vegada que es contracta un venedor a l'empresa, també hem de tenir l'opció de donar-lo de baixa quan aquest deixa de treballar a l'empresa.

Consulta i modificació

Dels venedors ens pot interessar consultar i/o modificar per exemple les seves dades de contacte.

Ordena i busca

Alhora de visualitzar el llistat de venedors igual que amb els productes i els clients, hem de tenir l'opció d'ordenar-los segons un criteri específic, en aquest cas, segons el DNI o el nom. O també ens pot interessar buscar un venedor introduint el seu DNI o nom directament, això ens ha de mostrar les dades del venedor que busquem en cas que existeixi.

Hi ha altres elements d'interès del domini d'aquest projecte com poden ser els catàlegs, les categories, les marques, les talles, els colors, els tipus d'iva

que no descriurem per no allargar massa la memòria i per ser molt simples i idèntics a lo explicat anteriorment.

3.4. Vendes

Cada cop que una persona realitza una compra a l'establiment, serà iniciada una venda. Aquesta consistirà en seleccionar des del catàleg els productes que es disposen a comprar o entrant el codi del producte directament, quedant indicat en una línia diferent per cada producte, la descripció d'aquest, la quantitat, el preu unitari i l'import total de la línia (quantitat x producte unitari), i si el producte disposa de talles i colors s'ha de tenir l'opció de seleccionar la talla i color del producte que es té per vendre. El preu unitari s'hauria de poder modificar a la mateixa línia de venda en cas de necessitat. Al final es calcularà automàticament la suma total dels imports. Cada venda s'assignarà al venedor que l'ha fet i al client, en cas que ho sigui, o altrament s'assignarà a una figura anomenada comprador.

Cada línia es podrà eliminar per contemplar casos d'error o de canvi de producte en l'últim moment per part del client. Finalment, el preu total de la venda s'hauria de calcular en funció del descompte que té el client, en cas que en tingui.

Un cop preparada la venda, hem de tenir tres opcions: cobrar-la, cancel·lar-la o passar-la al compte del client (en el cas que ho sigui). En cas de cobrar-la es guardaran les dades per tal de poder fer consultes o extreure informes. Si es cancel·la la venda en l'últim moment no es produeix cap canvi i simplement es tanca la finestra, i si indiquem que la volem traspasar al compte del client, s'emmagatzemarà considerant-la com a no cobrada.

Es podria donar la situació que un comprador habitual necessités deixar a deure una venda, pel que ha de ser necessàriament client, així que si es dóna el cas haurem de poder donar d'alta el client sense perdre les dades de la venda que ja està preparada.

3.5. Compres

Un cop rebuts els productes comprats s'hauria de fer l'ingrés d'aquests. Per tal d'iniciar el procés de compra s'haurien d'entrar els productes nous que no existeixen a la base de dades de l'aplicació, això es pot fer tant abans d'iniciar la compra com des de la mateixa pantalla de compra. Un cop entrats es comença la recepció de productes, de cada producte s'hauria d'enregistrar el codi d'aquest, la descripció, la quantitat, el preu de cost, el preu de venda, i el total per cada línia, el total de compra s'hauria de calcular automàticament al final de tot, en cas que d'un producte hi hagin diferents talles i colors, igual que en el procés de venda, el sistema ens ha de mostrar un llistat amb les diferents talles i colors del producte per poder fer la selecció.

Cada compra s'assignarà al venedor que l'ha rebut i al proveïdor que l'ha servit. Cada línia es podrà eliminar per contemplar casos d'error.

Un cop preparada la compra, hem de tenir dos opcions: acceptar-la o cancel·lar-la. En cas d'acceptar-la es guardaran les dades, i observacions en cas que n'hi hagi, per tal de poder fer consultes o extreure informes. Si es cancel·la la compra en l'últim moment no es produeix cap canvi i simplement es tanca la finestra,

Igual que les vendes es podria donar la situació que el proveïdor del qual estem fent l'ingrés dels seus productes no estigui entrant a la base de dades, així que si es dona el cas haurem de poder donar-lo d'alta perdre les dades de la compra que ja està preparada.

3.6. Llistats i resums estadístics

En aquest apartat es pretén treure llistats de diferents conceptes en funció de diferents criteris. Aquests conceptes són els productes, els clients, els Proveïdors, els venedors les vendes i les compres.

3.7. Manual d'usuari i còpies de seguretat

Per tal de completar el sistema, caldrà que es disposi d'un manual per a l'usuari amb la descripció del funcionament i també de l'opció de fer còpies de seguretat. Aquests no s'adjuntaran per no allargar la memòria, sinó que formaran part de la pròpia aplicació dins el menú.

4. Avaluació tecnològica

4.1. Llenguatges de programació

Avui en dia existeixen milers de llenguatges de programació, els quals van apareixent contínuament; és per això que ens limitarem a l'avaluació de només **C++** i **Java**, no per la seva qualitat o la seva potència, sinó per la seva popularitat.

sintaxi:

Pràcticament idèntica en C++ i Java.

Orientació a objectes:

En C++ es poden escriure programes orientats a objectes, no orientats a objectes, o barreja d'ambdós (per exemple, tenir classes amb funcions globals en el mateix programa).

Java és purament orientat a objectes; qualsevol funció ha de pertànyer a alguna classe (és un mètode). Hi ha la funció main, però és un mètode estàtic d'una classe (la classe principal).

Control sobre la plataforma (maquinari S.O.):

C++: el programador té control del processador.

Java: el programa s'executa sobre un processador virtual (JVM) que abstreu el programador de la plataforma concreta.

Codi de màquina generat:

C++: el compilador genera codi de màquina per a una plataforma Concreta.

Java: el compilador genera codi de màquina (bytecode) per al processador JVM; aquest codi ha de ser interpretat a posteriori.

Alliberament de Memòria:

C++: a càrrec del programador.

Java: automàtica, els objectes creats són destruïts automàticament. Un procés de la JVM anomenat recol·lector d'escombraries (garbage collector).

punters:

C++: hi ha punters amb una aritmètica ben definida, i entre altres coses, permeten al programador explorar la memòria del computador físic.

Java: no existeix el concepte de punter, els objectes són accedits mitjançant referències (una referència es pot considerar com el mateix propi identificador de l'objecte), no hi ha aritmètica de referències.

paquets:

C++: no existeix tal concepte; el més aproximat són els arxius de biblioteca (. h).

Java: un paquet és una entitat organitzativa que permet agrupar classes, interfícies i excepcions, així com que els seus elements disposin d'una determinada visibilitat pel que fa a elements externs al paquet.

Herència múltiple:

C++: hi ha.

Java: no existeix per considerar poc eficient (això va implicar algunes crítiques a Java des de certs sectors).

compiladors:

C++: Molts compiladors. Poden ser bastant diferents.

Java: Un únic compilador sempre igual.

objectes:

C++: Cap ancestre comú.

Java: Tots els objectes hereten automàticament de la classe Object.

Elecció del llenguatge de programació

Per fer l'elecció del llenguatge de programació necessitem que sigui un llenguatge senzill i entenedor. Volem treballar amb orientació a objectes i sobretot que faciliti la creació de la interfície gràfica.

Personalment el llenguatge Java l'havia fet servir molt durant la meua carrera a diferència del C++, amb el que he treballat molt poc, tenint en compte la voluntat d'aprofundir amb aquest llenguatge i que C++ també compleix les condicions s'escull el C++, i concretament el compilador C++ Builder 6.

4.2. SGBD

Els sistemes de gestió de bases de dades són un tipus de programari molt específic, dedicat a servir d'interfície entre la base de dades, l'usuari i les aplicacions que la utilitzen.

El propòsit general dels sistemes de gestió de bases de dades és el de gestionar de manera clara, senzilla i ordenada un conjunt de dades que posteriorment es convertiran en informació rellevant per a una organització.

Una de les condicions primordials de cara a l'elecció del sistema gestor de la base de dades és que sigui lliure. Per tant, en aquest cas limitarem la cerca a dos tipus de SGBD: **MySQL i PostgreSQL**.

Elecció del SGBD

La base de dades de l'aplicació no requereix una gran potència, ja que es tracta d'una base que està pensada per a gestionar una sola caixa, però sí que necessitarem que el sistema sigui capaç de funcionar de manera eficient.

D'entrada hem decidit seleccionar un SGBD lliure. Els eines que ens ofereixen els dos SGBD's són més que suficients per a la nostra aplicació, tant pel que fa als tipus de dades suportades, així com als operadors inclosos en el SGBD, la connexió des de PHP, les prestacions en la creació de les estructures (taules, índexs...) i les consultes.

La diferència més important que fa que el SGBD elegit sigui MySQL és que el PostgreSQL permet treballar directament amb objectes, i el disseny d'objectes fet pel programa no contempla objectes estructurats i per tant es poden guardar directament amb taules relacionals. A més MySQL ens

permet documentació molt extensa de suport tant en la seva pàgina web en diverses idiomes, com la d'optar per consultar l'amplíssima gamma d'usuaris del producte.

4.3. Seguretat

La base principal d'aquest projecte gira al voltant de les dades personals.

És per això que s'ha de seguir molt acuradament tota la legislació vigent LOPD (Llei Orgànica de Protecció de Dades) per poder-ne assegurar el compliment. Aquesta Llei té com a objecte garantir i protegir, pel que fa al tractament de les dades personals, les llibertats públiques i els drets fonamentals de les persones físiques, i especialment del seu honor i la seva intimitat personal i familiar.

De tot això en podem concloure que per tal de complir amb els articles 5 i 6 referents al dret de la informació en la recollida de dades i el consentiment, haurem de fer signar als clients la conformitat amb que recollim, emmagatzemem i tractem les seves dades personals.

5. Prototip 1: catàleg de productes

5.1. Requeriments

5.1.1. Catàleg

- El sistema ha de permetre que l'actor pugui donar d'alta un nou catàleg.
- El sistema ha de permetre que l'actor pugui editar les característiques dels catàlegs existents.
- El sistema ha de permetre ordenar les dades dels catàlegs disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa un catàleg en cas que no hi hagi cap producte ni categoria que hi pertanyin.

- El sistema ha de permetre que l'actor pugui buscar un catàleg en concret.

5.1.2. Categoria

- El sistema ha de permetre que l'actor pugui donar d'alta noves categories en qualsevol dels catàlegs disponibles.
- El sistema ha de permetre que l'actor pugui editar les característiques de les categories existents.
- El sistema ha de permetre ordenar les dades de les categories disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa una categoria en cas que no hi hagi cap producte que hi pertanyi.
- El sistema ha de permetre que l'actor pugui buscar una categoria en concret.

5.1.3. Marca

- El sistema ha de permetre que l'actor pugui donar d'alta una nova marca.
- El sistema ha de permetre que l'actor pugui editar les característiques de les marques existents.
- El sistema ha de permetre ordenar les dades de les marques disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa una marca en cas que no hi hagi cap producte que hi pertanyi.
- El sistema ha de permetre que l'actor pugui buscar una marca en concret.

5.1.4. Color

- El sistema ha de permetre que l'actor pugui donar d'alta un nou color.

- El sistema ha de permetre que l'actor pugui editar les característiques dels colors existents.
- El sistema ha de permetre ordenar les dades dels colors disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa un color en cas que no hi hagi cap producte d'aquest color.
- El sistema ha de permetre que l'actor pugui buscar un color en concret.

5.1.5. Talla

- El sistema ha de permetre que l'actor pugui donar d'alta un nova talla.
- El sistema ha de permetre que l'actor pugui editar les característiques de les talles existents.
- El sistema ha de permetre ordenar les dades de les talles disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa una talla en cas que no hi hagi cap producte d'aquesta talla.
- El sistema ha de permetre que l'actor pugui buscar una talla en concret.

5.1.6. Iva

- El sistema ha de permetre que l'actor pugui donar d'alta un nou tipus d'iva.
- El sistema ha de permetre que l'actor pugui editar les característiques dels diferents tipus d'iva.
- El sistema ha de permetre ordenar els tipus d'iva disponibles segons el camp especificat.

- El sistema ha de permetre que l'actor pugui donar de baixa un tipus d'iva en cas que no hi hagi cap producte que el tingui associat.

5.1.7. Proveïdor

- El sistema ha de permetre que l'actor pugui donar d'alta un nou proveïdor.
- El sistema ha de permetre que l'actor pugui editar les característiques dels proveïdors existents.
- El sistema ha de permetre ordenar les dades dels proveïdors disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa un proveïdor en cas que no hi hagin productes donats d'alta que serveixi aquest proveïdor, i no hi hagi cap pantalla de compra d'aquest proveïdor pendent de tancar.
- El sistema ha de permetre que l'actor pugui buscar un proveïdor en concret.

5.1.8. Producte

- El sistema ha de permetre que l'actor pugui donar d'alta un nou producte en qualsevol dels catàlegs i categories disponibles.
- El sistema ha de permetre que l'actor pugui editar les característiques dels productes existents.
- El sistema ha de permetre ordenar les dades dels productes disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa un producte en cas que no tingui estoc i no surti en cap pantalla de venda o compra pendent de tancar.
- El sistema ha de permetre exportar les dades dels productes disponibles a un arxiu d'Excel.

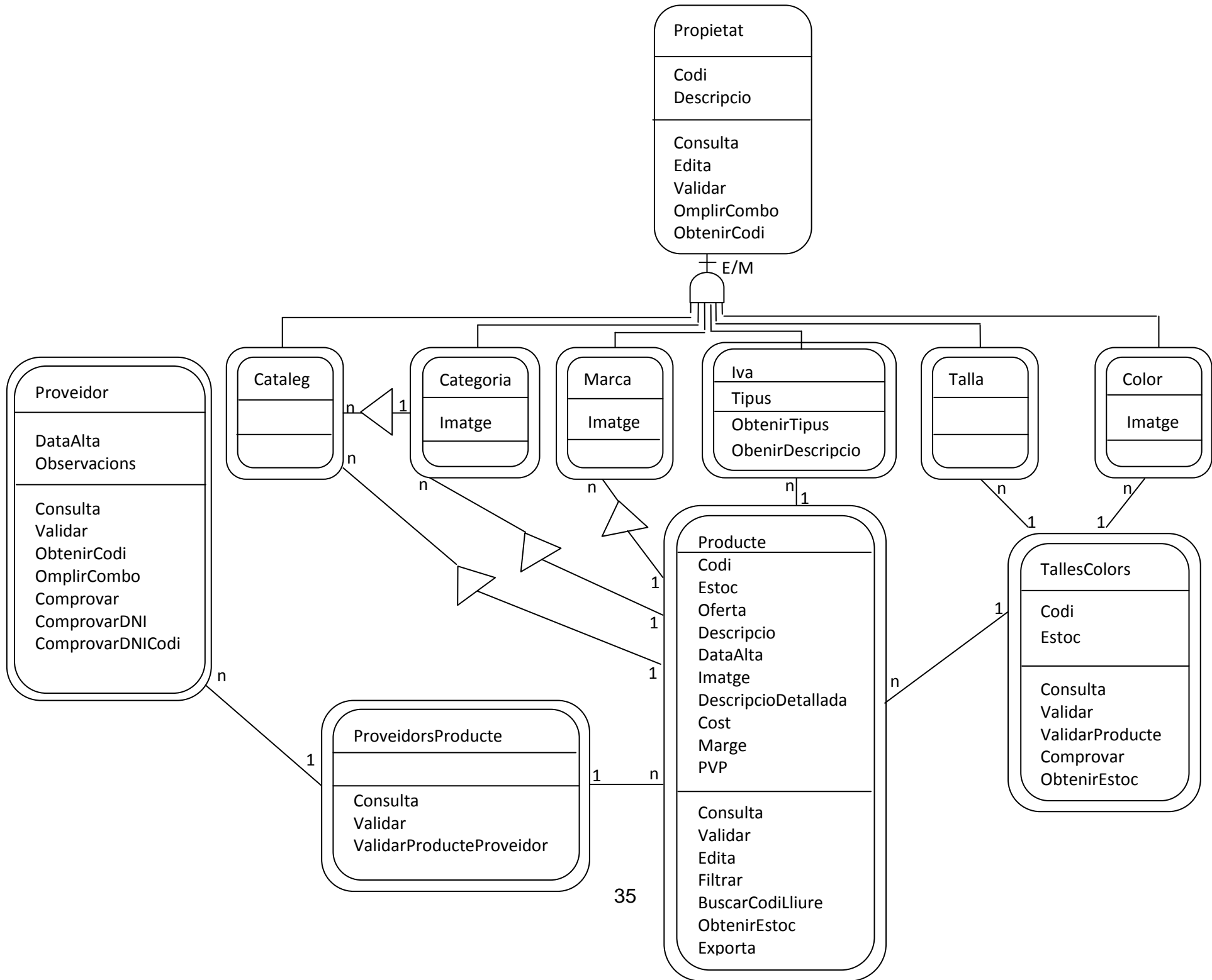
- El sistema ha de permetre que l'actor pugui filtrar l' informació segons diferents criteris (catàleg, categoria, marca....).
- El sistema ha de permetre que l'actor pugui buscar un producte en concret.
- El sistema ha de permetre que l'actor pugui assignar combinacions de talles i colors a un producte.
- El sistema ha de permetre que l'actor pugui assignar un proveïdor a un producte.
- El sistema ha de permetre visualitzar les diferents combinacions de talles i colors d'un producte.
- El sistema ha de permetre visualitzar els diferents proveïdors que serveixen un producte.

5.2. Anàlisi

El primer pas serà la realització del diagrama de classes, que consisteix en el diagrama principal per al modelatge orientat a objectes. Representa les classes del sistema juntament amb les seves relacions estructurals, agregació i associació. També es definiran els atributs i operacions.

Amb el diagrama de classes entendrem quines funcions poden ser dutes a terme, com es poden desenvolupar i quins tipus d'objectes interactuen entre si.

Degut a que el diagrama de classes és molt gran, només mostrarem els objectes que fan referència a cada apartat, A continuació mostrarem el diagrama de classes del catàleg de productes i els objectes analitzats en aquest apartat.



Per no ampliar gaire la memòria, només es farà l'especificació d'alguns serveis perquè hi ha casos en els que el comportament dels serveis és molt semblant per això només detallarem alguns sinó seria molt repetitiu.

Producte

Codi: identificador del producte; INTEGER.

Descripció: descripció del producte; VARCHAR.

Estoc: estoc del producte; INTEGER.

Cost: preu de cost del producte; DOUBLE.

Marge: marge de guanys del producte; DOUBLE.

PVP: preu de venda públic del producte; DOUBLE.

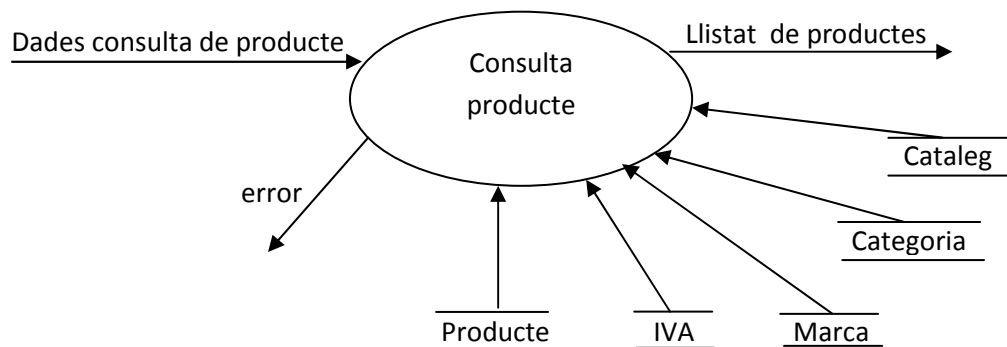
DataAlta: data d'alta del producte; DATE.

Imatge: nom de la foto del producte; VARCHAR.

DescripcioDetallada: descripció detallada del producte; VARCHAR.

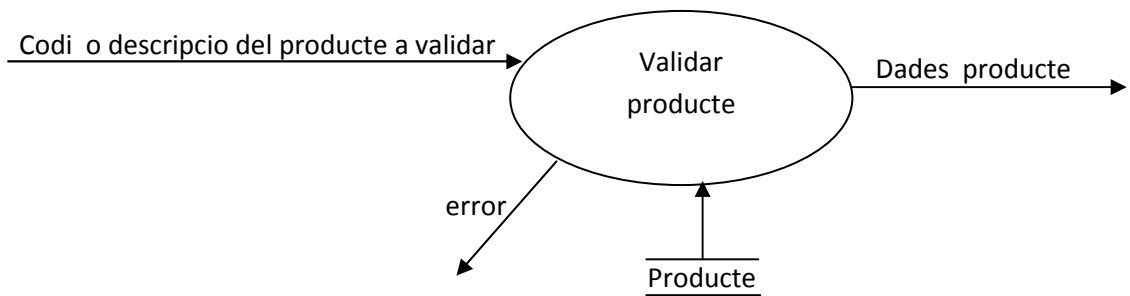
Oferta: indica si el producte esta d'oferta o no; VARCHAR.

Consulta: mostra un llistat amb les dades dels productes de forma detallada ordenat per la descripció.



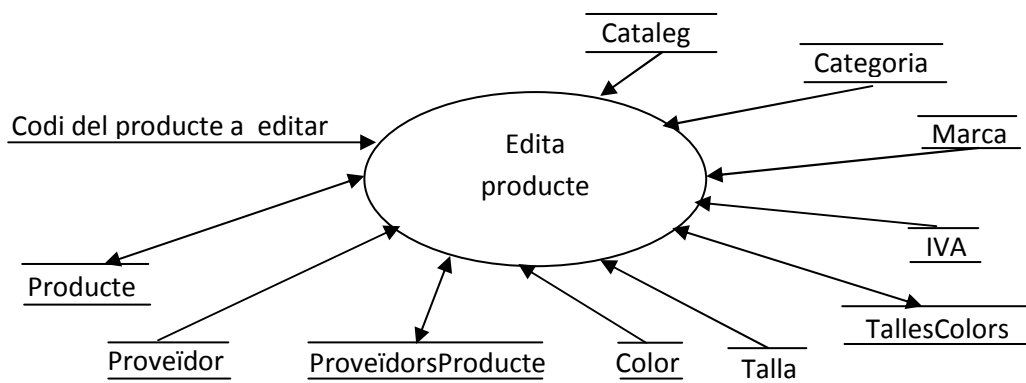
DFD de la consulta de productes

Validar: Busca un producte donat el seu codi o la descripció.



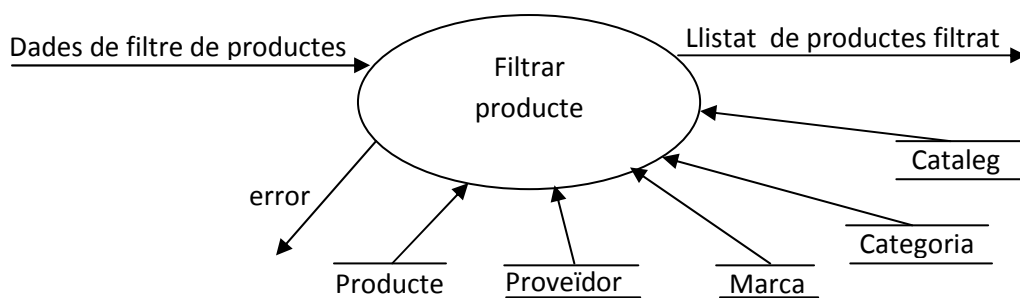
DFD de validar producte

Edita: Edita les dades d'un producte donat el seu codi.



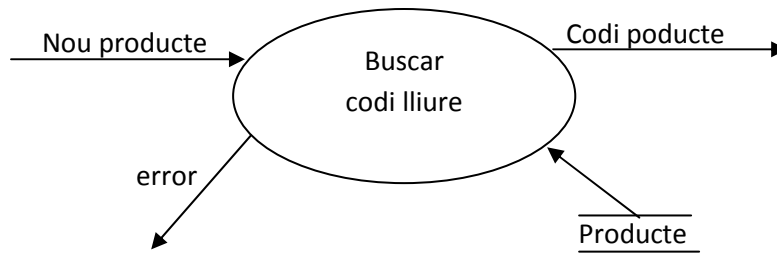
DFD d'edita producte

Filtrar: mostra un llistat filtrat per proveïdor, marca, catàleg, categoria, estoc i oferta amb les dades dels productes de forma detallada ordenats per la descripció.



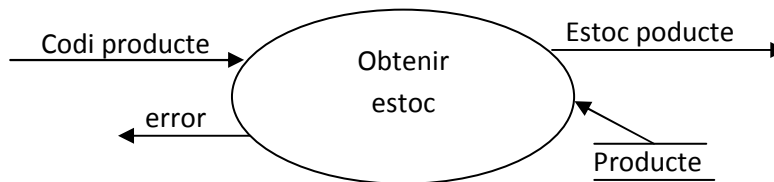
DFD de filtrar producte

BuscarCodiLliure: Busca un Codi lliure per assignar al producte.



DFD de buscar el codi lliure del producte

ObtenirEstoc: donat el codi d'un producte ens retorna l'estoc.



DFD d'obtenir l'estoc del producte

Propietat

Codi: identificador de la Propietat; INTEGER.

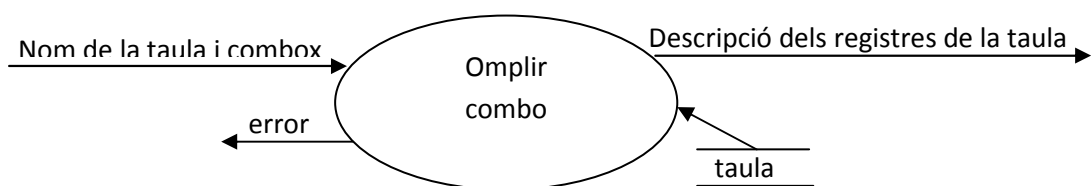
Descripció: descripció de la propietat; VARCHAR.

Consulta: Donada una taula, es mostra un llistat amb les dades de la taula de forma detallada ordenat per descripció.

Edita: Edita les dades d'un registre de la taula donat el seu codi.

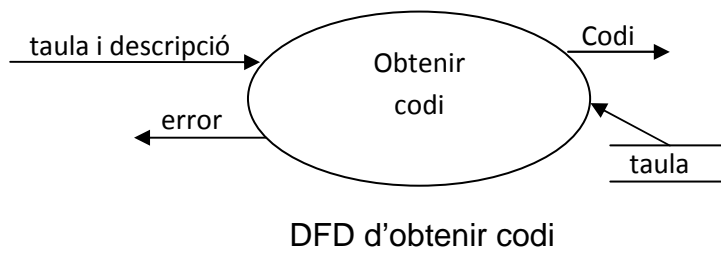
Validar: Busca un registre de la taula donat el seu codi.

OmplirCombo: donada una taula, es mostren els seus registres amb un ComboBox.



DFD d'obtenir l'estoc del producte

ObtenirCodi: donada la descripció i la taula, retorna el codi del registre de la taula amb la descripció donada.



Categoria

Imatge: nom d'imatge de la categoria; VARCHAR.

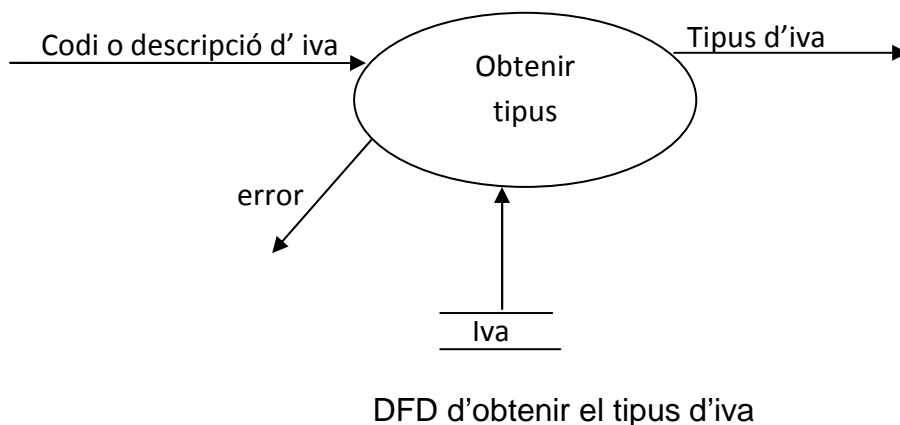
Marca

Imatge: nom d'imatge de la marca; VARCHAR.

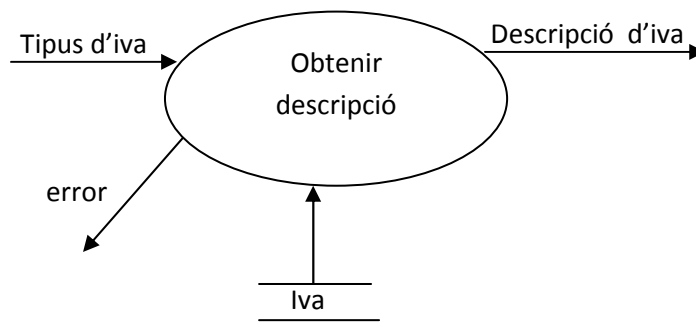
Iva

Tipus: tipus d'iva; INTEGER.

ObtenirTipus: Donat el codi o la descripció d'un tipus d'iva ens retorna el tipus d'iva.



ObtenirDescripcio: Donat el tipus d'un tipus d'iva ens retorna la descripció.



DFD d'obtenir la descripció del tipus d'iva

Color

Imatge: nom d'imatge del color; VARCHAR.

TallesColors

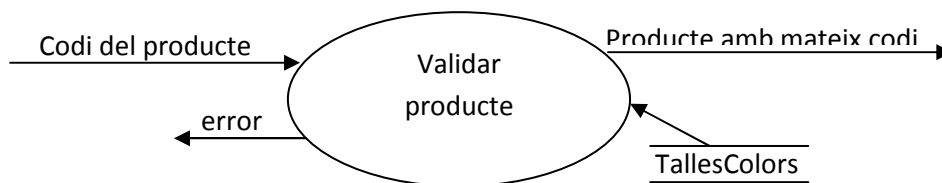
Codi: identificador de Talles Colors; INTEGER.

Estoc: Estoc del producte; INTEGER.

Consulta: Donat un producte, es mostra un llistat de les diferents combinacions de les talles i colors d'aquest producte amb el seu estoc corresponent.

Validar: Busca un producte amb un color i una talla, donats el codi del producte, el codi del color i el codi de la talla.

ValidarProducte: Donat un producte, comprova si té talles i colors.



DFD de validar producte

Comprovar: Donat un color i una talla, mirar si consten amb un StringGrid.

ObtenirEstoc: donat un producte, un color i una talla, retorna l'estoc d'aquest producte amb aquest color i aquesta talla.

Proveïdor

DataAlta: data d'alta del proveïdor; DATE.

Observacions: observacions sobre el proveïdor; VARCHAR.

Els altres atributs estan a empresa.

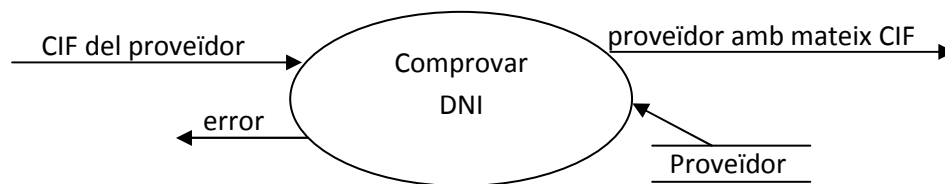
Consulta: mostra un llistat amb les dades dels proveïdors de forma detallada ordenat per la descripció.

Validar: Busca un proveïdor donat el seu codi.

ObtenirCodi: donada la raó social, retorna el codi del proveïdor.

OmplirCombo: es mostra la raó social dels proveïdors entrats a la base de dades amb un ComboBox.

ComprovarDNI: Donat el CIF d'un proveïdor, comprova si hi ha algun proveïdor entrat a base de dades que tingui el mateix CIF.



DFD de comprovar DNI

Comprovar: Comprova si un proveïdor existeix amb un StringGrid.

ComprovarDNICodi: Donat el CIF i el codi d'un proveïdor, comprova si hi ha algun proveïdor amb el mateix CIF i codi diferent.

ProveïdorsProducte

Consulta: donat un producte, mostra el llistat de proveïdors del producte.

Validar: busca el proveïdor d'un producte, donat el codi del producte i el codi del proveïdor.

ValidarProducteProveïdor: donat un producte mirar si té proveïdors assignats, o al revés.

5.3. Disseny del sistema

5.3.1. Interfícies

Per tal d'assegurar un bon disseny i mantenir una coherència visual en les interfícies amb l'usuari, s'ha tingut en compte una sèrie de característiques:

Mínim esforç: en totes les finestres, com ja hem comentat en l'apartat d'anàlisi del sistema, es segueix un mateix model perquè l'usuari s'hi pugui adaptar fàcilment.

Mínima Memòria: s'ha simplificat el sistema de tal manera que només s'han utilitzat les eines mínimes i necessàries per fer-ho anar, perquè hi hagi una comunicació senzilla entre l'usuari i el programa.

Notificació de problemes d'immediat: si hi ha alguna acció que no pot ser realitzada es mostra per pantalla un missatge sobre com solucionar-ho.

Ajuda: s'ha implementat un manual d'usuari com a ajuda de cara a l'usuari.

Tolerància en errors: en el cas d'haver-hi un error, permet abandonar l'acció i solucionar-lo.

Confirmació en accions destructives: abans d'eliminar qualsevol objecte, s'avisa a l'usuari amb un missatge que surt per pantalla per tal de confirmar-ho o cancel·lar en cas d'error.

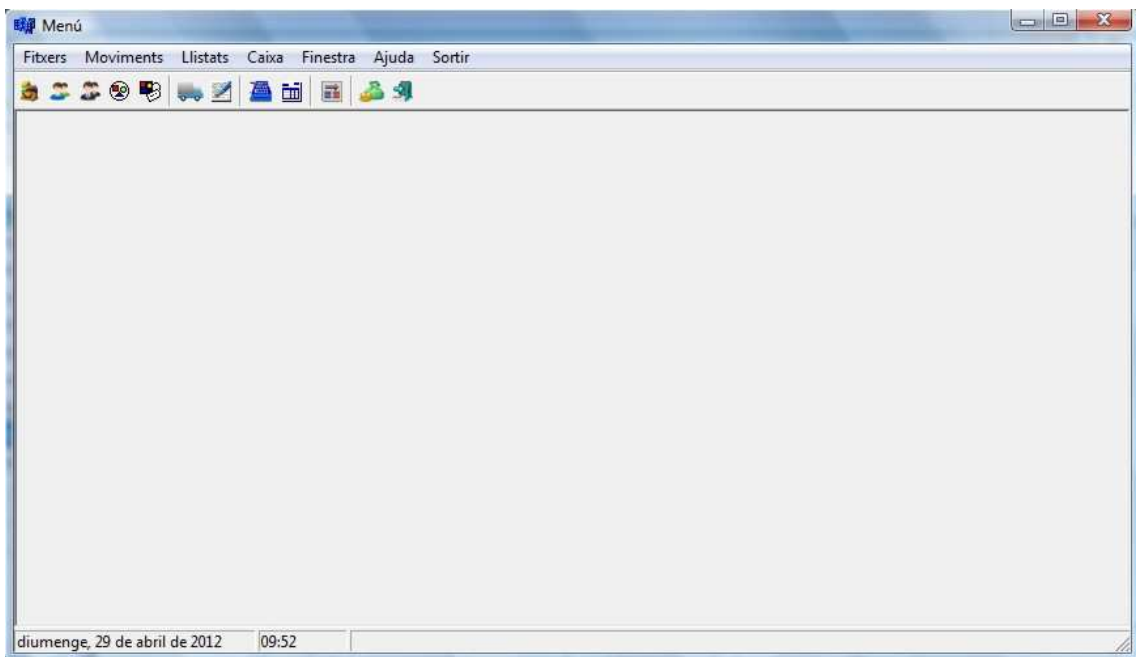
Mostrar informació: Només mostrar l'informació rellevant.

Entrada de dades: Pel que fa a l'entrada de dades, tindran el següent comportament:

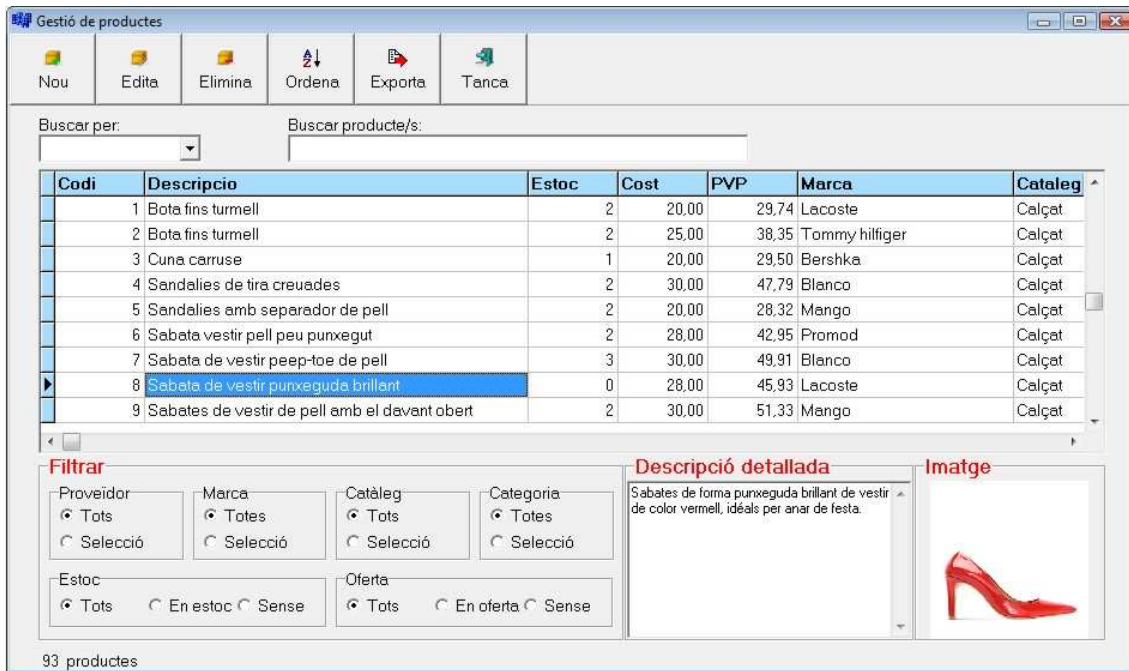
- Quan les condicions necessàries per poder dur a terme les accions no s'hagin complert, els botons estaran deshabilitats.

- En el cas que l'usuari vulgui donar d'alta un registre nou i algun atribut d'aquest estigui relacionat amb una dada existent a la base de dades, apareixerà un ComboBox que contindrà totes les possibles opcions.

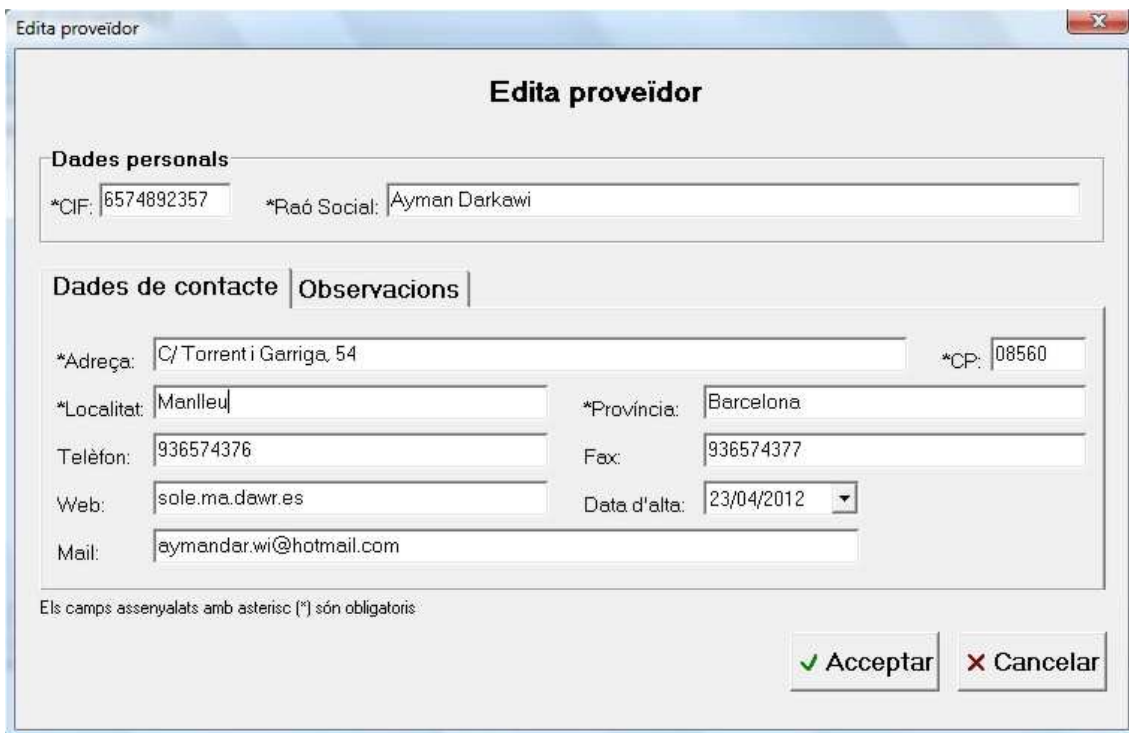
Seguidament, es mostraran algunes d'interfícies per exemplificar les característiques esmentades anteriorment:



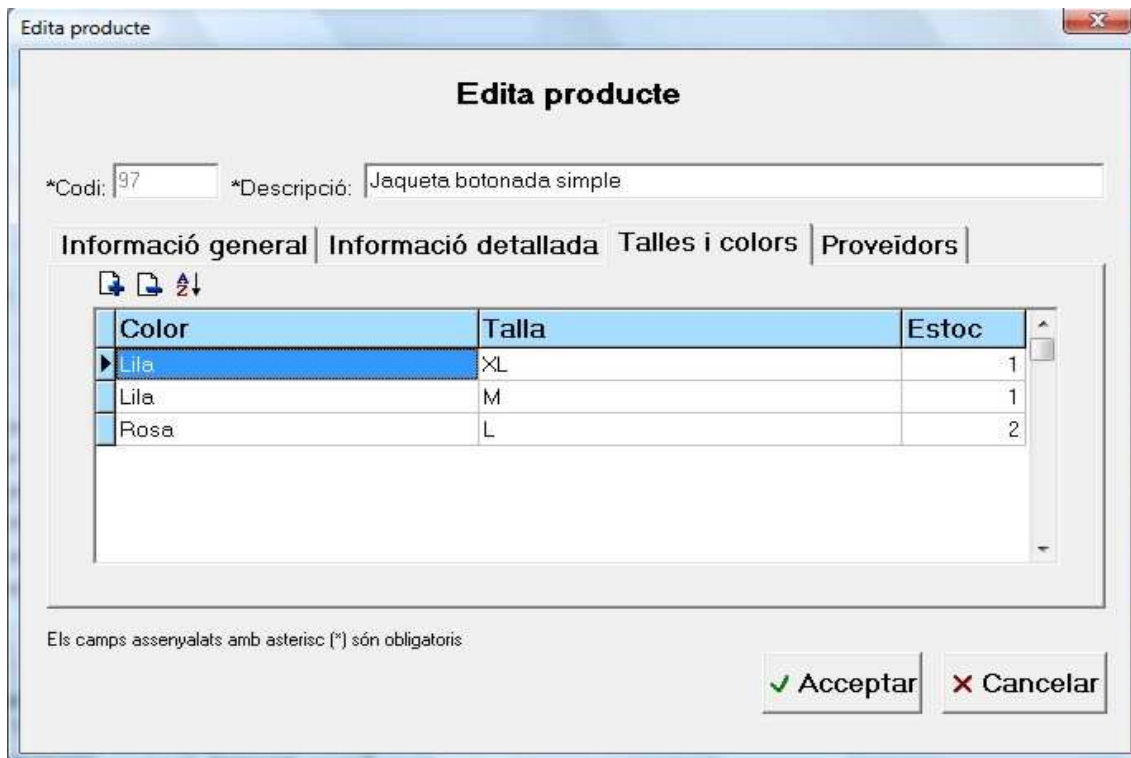
Pantalla del menú principal



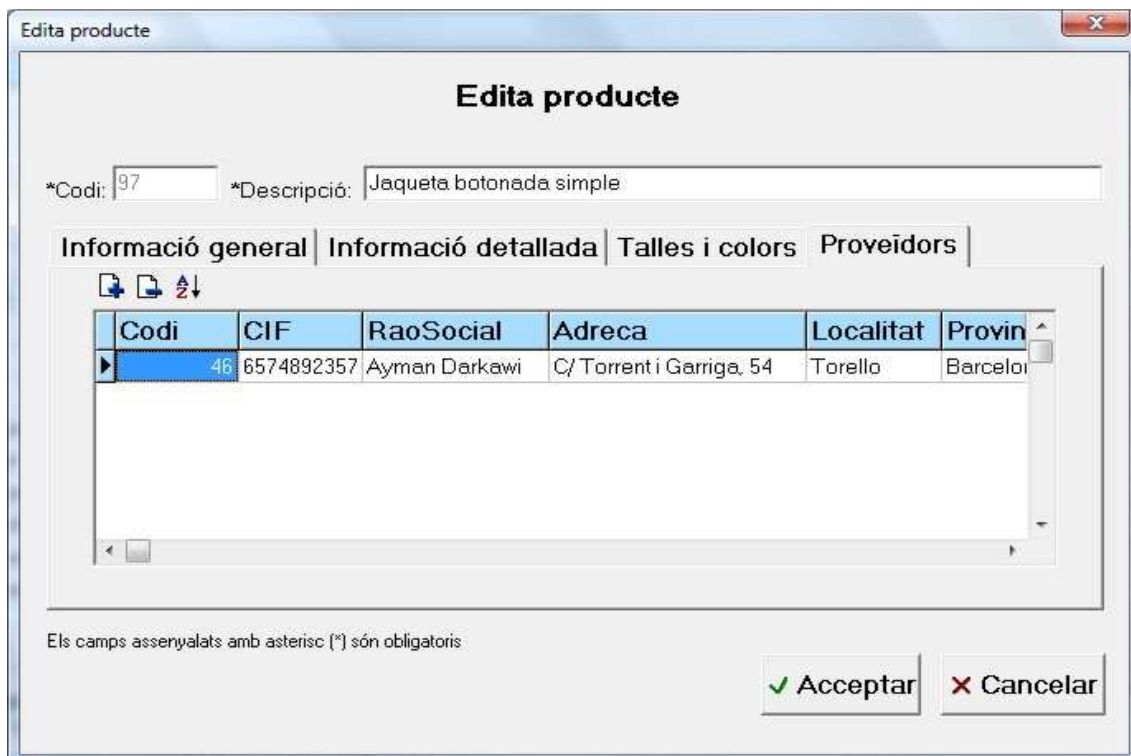
Pantalla de la gestió de productes



Pantalla d'editar proveïdor



Pantalla d'editar producte que mostra les combinacions de talles i colors



Pantalla d'editar de producte que mostra els proveïdors del producte

Seguidament, es mostrarà l'ampliació del diagrama de classe anterior.

5.3.2. Esdeveniments

NOM ESDEVENIMENT: Gestió de productes

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels productes.

DESCRIPCIÓ: es busquen les dades dels productes amb les descripcions del catàleg, la categoria, la marca i el tipus d'iva al que pertany cada producte i es mostren per pantalla.

SERVEIS: Producte.consulta, Cataleg.consulta, Categoria.consulta, Marca.consulta, Iva.consulta.

NOM ESDEVENIMENT: Nou producte

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades del nou producte.

DESCRIPCIÓ: se selecciona el catàleg, la categoria, la marca i el tipus d'iva, s'entren les dades del producte, les combinacions de talles i colors i la llista de proveïdors si en té.

SERVEIS: Producte.consulta, Cataleg.consulta, Categoria.consulta, Marca.consulta, Iva.consulta, Producte.alta, TallesColors.alta, ProveïdorsProducte.alta.

NOM ESDEVENIMENT: Edita producte

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades del producte.

DESCRIPCIÓ: se selecciona el catàleg, la categoria, la marca i el tipus d'iva, es modifiquen les dades del producte i s'actualitzen les combinacions de talles i colors i la llista de proveïdors si en té.

SERVEIS: Producte.consulta, Cataleg.consulta, Categoria.consulta, Marca.consulta, Iva.consulta, TallesColors.consulta, Talla.consulta, Color.Consulta, ProveidorsProducte.consulta, Proveidor.consulta, Producte.edita, TallesColors.baixa, ProveidorsProducte.baixa.

NOM ESDEVENIMENT: Gestió de catàlegs

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels catàlegs.

DESCRIPCIÓ: es busquen les dades dels catàlegs i es mostren per pantalla.

SERVEIS: Cataleg.consulta.

NOM ESDEVENIMENT: Nou catàleg

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades del nou catàleg.

DESCRIPCIÓ: s'entra la descripció del nou catàleg i es guarda a la base de dades.

SERVEIS: Cataleg.consulta, Cataleg.alta.

NOM ESDEVENIMENT: Edita catàleg

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades del catàleg.

DESCRIPCIÓ: es modifica la descripció del catàleg i es guarda a la base de dades.

SERVEIS: Cataleg.consulta, Cataleg.edita.

NOM ESDEVENIMENT: Gestió de categories

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de les categories.

DESCRIPCIÓ: es busquen les dades de les categories i el catàleg que pertany cada categoria i es mostren per pantalla.

SERVEIS: Categoria.consulta, Cataleg.consulta.

NOM ESDEVENIMENT: Nova categoria

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades de la nova categoria.

DESCRIPCIÓ: se selecciona el catàleg i s'entra la descripció de la nova categoria i es guarda a la base de dades.

SERVEIS: Categoria.consulta, Cataleg.consulta, Categoria.alta.

NOM ESDEVENIMENT: Edita categoria

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades de la categoria.

DESCRIPCIÓ: se selecciona el catàleg i es modifica la descripció de la categoria i es guarda a la base de dades.

SERVEIS: Categoria.consulta, Cataleg.consulta, Categoria.edita.

NOM ESDEVENIMENT: Gestió de marques

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de les marques.

DESCRIPCIÓ: es busquen les dades de les marques i es mostren per pantalla.

SERVEIS: Marca.consulta.

NOM ESDEVENIMENT: Nova marca

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades de la nova marca.

DESCRIPCIÓ: s'entra la descripció de la nova marca i es guarda a la base de dades.

SERVEIS: Marca.consulta, Marca.alta.

NOM ESDEVENIMENT: Edita marca

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades de la marca.

DESCRIPCIÓ: es modifica la descripció de la marca i es guarda a la base de dades.

SERVEIS: Marca.consulta, Marca.edita.

NOM ESDEVENIMENT: Gestió d'iva

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels tipus d'iva.

DESCRIPCIÓ: es busquen els diferents tipus d'iva amb les seves dades i es mostren per pantalla.

SERVEIS: IVA.consulta.

NOM ESDEVENIMENT: Nou iva

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades del nou tipus d'iva.

DESCRIPCIÓ: s'entra el tipus d'iva i la descripció i es guarda a la base de dades.

SERVEIS: IVA.consulta, IVA.alta.

NOM ESDEVENIMENT: Edita IVA

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades del tipus d'iva.

DESCRIPCIÓ: es modifica el tipus d'iva i la descripció i es guarda a la base de dades.

SERVEIS: IVA.consulta, IVA.edita.

NOM ESDEVENIMENT: Gestió de Talles

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de les talles.

DESCRIPCIÓ: es busquen les dades de les talles i es mostren per pantalla.

SERVEIS: Talla.consulta.

NOM ESDEVENIMENT: Nova Talla

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades de la nova talla.

DESCRIPCIÓ: s'entra la descripció de la talla i es guarda a la base de dades.

SERVEIS: Talla.consulta, Talla.alta.

NOM ESDEVENIMENT: Edita Talla

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades de la talla.

DESCRIPCIÓ: es modifica la descripció de la talla i s'actualitza a la base de dades.

SERVEIS: Talla.consulta, Talla.edita.

NOM ESDEVENIMENT: Gestió de colors

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels colors.

DESCRIPCIÓ: es busquen les dades dels colors i es mostren per pantalla.

SERVEIS: Color.consulta.

NOM ESDEVENIMENT: Nou color

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades del nou color.

DESCRIPCIÓ: s'entra la descripció del color i es guarda a la base de dades.

SERVEIS: Color.consulta, Color.alta.

NOM ESDEVENIMENT: Edita color

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades del color.

DESCRIPCIÓ: es modifica la descripció del color i s'actualitza a la base de dades.

SERVEIS: Color.consulta, Color.edita.

NOM ESDEVENIMENT: Gestió de proveïdors

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels proveïdors.

DESCRIPCIÓ: es busquen les dades dels proveïdors i es mostren per pantalla.

SERVEIS: Proveidor.consulta.

NOM ESDEVENIMENT: Nou proveïdor

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades del nou proveïdor.

DESCRIPCIÓ: s'entren les dades del nou proveïdor i es guarda a la base de dades.

SERVEIS: Proveidor.consulta, Proveidor.alta.

NOM ESDEVENIMENT: Edita proveïdor

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades del proveïdor.

DESCRIPCIÓ: es modifiquen les dades del proveïdor i s'actualitzen a la base de dades.

SERVEIS: Proveidor.consulta, Proveidor.edita.

NOM ESDEVENIMENT: Seleccionar proveïdors

TIPUS ESDEVENIMENT: informació

RESPOSTA: es mostraran les dades dels proveïdors i es registrarà el proveïdor de la línia seleccionada als proveïdors del producte.

DESCRIPCIÓ: després de buscar les dades dels proveïdors i mostrar-les per pantalla, s'afegeix el proveïdor seleccionat als proveïdors del producte.

SERVEIS: Proveidor.consulta, ProveidorsProducte.alta.

NOM ESDEVENIMENT: Seleccionar Talles i color

TIPUS ESDEVENIMENT: informació

RESPOSTA: es mostraran les dades de les talles i els colors i es registrarà la combinació de talla i color que s'hagi seleccionat a les talles i colors del producte.

DESCRIPCIÓ: després de buscar les dades de les talles i colors i mostrar-les per pantalla, s'afegeix la combinació de talla i color seleccionats a les talles i colors del producte.

SERVEIS: Talla.consulta, Color.consulta, TallesColors.alta.

NOM ESDEVENIMENT: Seleccionar proveïdors

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels proveïdors i s'afegirà el proveïdor de la línia seleccionada al StringGrid de la pantalla de proveïdors del producte.

DESCRIPCIÓ: després de buscar les dades dels proveïdors i mostrar-les per pantalla, s'afegeix el proveïdor seleccionat al StringGrid de la pantalla de proveïdors del producte.

SERVEIS: Proveïdor.consulta.

NOM ESDEVENIMENT: Seleccionar Talles i color

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de les talles i els colors i es registrarà la combinació de talla i color que s'hagi seleccionat al StringGrid de la pantalla de talles i colors del producte.

DESCRIPCIÓ: després de buscar les dades de les talles i els colors i mostrar-les per pantalla, s'afegeix la combinació de talla i color seleccionats al StringGrid de la pantalla de talles i colors del producte.

SERVEIS: Talla.consulta, Color.consulta.

5.4. Disseny de la base de dades

```
CREATE TABLE `producte` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Descripcio` varchar(45) NOT NULL,  
  `Cataleg` int(10) unsigned NOT NULL,  
  `Categoria` int(10) unsigned NOT NULL,  
  `Marca` int(10) unsigned NOT NULL,  
  `IVA` int(10) unsigned NOT NULL,  
  `DataAlta` timestamp NOT NULL default CURRENT_TIMESTAMP on  
update CURRENT_TIMESTAMP,
```

```
`Imatge` varchar(250) default NULL,  
`DescripcioDetallada` varchar(1000) default NULL,  
`Estoc` int(11) NOT NULL,  
`Cost` double NOT NULL default '0',  
`Marge` double NOT NULL default '0',  
`PVP` double NOT NULL,  
`Oferta` int(10) unsigned NOT NULL default '0',  
PRIMARY KEY (`Codi`),  
)
```

```
CREATE TABLE `cataleg` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Descripcio` varchar(45) NOT NULL,  
  PRIMARY KEY (`Codi`)  
)
```

```
CREATE TABLE `categoria` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Descripcio` varchar(45) NOT NULL,  
  `Cataleg` int(10) unsigned NOT NULL,  
  `Imatge` varchar(250) default NULL,  
  PRIMARY KEY (`Codi`),  
)
```

```
CREATE TABLE `marca` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Descripcio` varchar(45) NOT NULL,  
  `Imatge` varchar(250) NOT NULL,  
  PRIMARY KEY (`Codi`)  
)
```

```
CREATE TABLE `iva` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Descripcio` varchar(45) NOT NULL,
```



```
`Tipus` varchar(5) NOT NULL,  
PRIMARY KEY (`Codi`)  
)
```

```
CREATE TABLE `talla` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Descripcio` varchar(45) NOT NULL,  
  PRIMARY KEY (`Codi`)  
)
```

```
CREATE TABLE `color` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Descripcio` varchar(45) NOT NULL,  
  `Imatge` varchar(250) NOT NULL,  
  PRIMARY KEY (`Codi`)  
)
```

```
CREATE TABLE `tallescors` (  
  `Talla` int(10) unsigned NOT NULL,  
  `Color` int(10) unsigned NOT NULL,  
  `Estoc` int(11) NOT NULL,  
  `Producte` int(10) unsigned NOT NULL,  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  PRIMARY KEY USING BTREE (`Codi`),  
)
```

```
CREATE TABLE `proveidor` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `CIF` varchar(10) character set latin1 NOT NULL,  
  `RaoSocial` varchar(45) character set latin1 NOT NULL,  
  `Adreca` varchar(50) character set latin1 NOT NULL,  
  `CP` varchar(10) character set latin1 NOT NULL,  
  `Localitat` varchar(50) character set latin1 NOT NULL,  
  `Provincia` varchar(50) character set latin1 NOT NULL,
```

```

`Telefon` varchar(15) character set latin1 NOT NULL,
`Fax` varchar(15) character set latin1 NOT NULL,
`Web` varchar(50) character set latin1 NOT NULL,
`DataAlta` date NOT NULL,
`Mail` varchar(50) character set latin1 NOT NULL,
`Observacions` varchar(1000) character set latin1 default NULL,
PRIMARY KEY (`Codi`)
)

```

```

CREATE TABLE `proveidorsproducte` (
  `Proveidor` int(10) unsigned NOT NULL,
  `Producte` int(10) unsigned NOT NULL,
  PRIMARY KEY USING BTREE (`Proveidor`,`Producte`),
)

```

5.5. Disseny del programa

- **Exemple d'exportar**

```

void TfGestioProductes::exportar()
{
    fGestioProductes->Panel3->Visible=true;
    Variant XL,v0,v1,v2,v3; //declare variables
    XL=Variant::CreateObject("excel.application"); //open excel application
    v0=XL.OlePropertyGet("Workbooks"); //Get workbook
    v0.OleProcedure("Add"); //add one work book
    v0=v0.OlePropertyGet("Item",1); //select workbook number 1
    v0=v0.OlePropertyGet("WorkSheets"); //Get worksheet
    v0.OleFunction("Add"); // add one work sheet
    v0=v0.OlePropertyGet("Item",1); //select worksheet number 1
    v1 = v0.OlePropertyGet("Cells"); //get a cell
    fGestioProductes->ProgressBar1->Min = 0;
    fGestioProductes->ProgressBar1->Position = 0;
    long lnContador = 1;
    //Se pinta el encabezado

```

```

v1.OlePropertyGet("Item",InContador,1).OlePropertySet("Value","Codi");

v1.OlePropertyGet("Item",InContador,2).OlePropertySet("Value","Descripcio
");

v1.OlePropertyGet("Item",InContador,3).OlePropertySet("Value","Estoc");

v1.OlePropertyGet("Item",InContador,4).OlePropertySet("Value","Cost");

v1.OlePropertyGet("Item",InContador,5).OlePropertySet("Value","PVP");

v1.OlePropertyGet("Item",InContador,6).OlePropertySet("Value","Marca");

v1.OlePropertyGet("Item",InContador,7).OlePropertySet("Value","Catalog");

v1.OlePropertyGet("Item",InContador,8).OlePropertySet("Value","Categoria"
);

v1.OlePropertyGet("Item",InContador,9).OlePropertySet("Value","IVA");

v1.OlePropertyGet("Item",InContador,10).OlePropertySet("Value","Marge");

v1.OlePropertyGet("Item",InContador,11).OlePropertySet("Value","DataAlta"
);

v1.OlePropertyGet("Item",InContador,12).OlePropertySet("Value","Imatge");

v1.OlePropertyGet("Item",InContador,13).OlePropertySet("Value","Oferta");

```

```

InContador++; //Incrementamos el renglon en excel

```

```

int InColor=40;
DBGrid1->DataSource->DataSet->First();
fGestioProductes->ProgressBar1->Max = DBGrid1->DataSource-
>DataSet->RecordCount;

```

```

while(!DBGrid1->DataSource->DataSet->Eof)
{
    fGestioProductes->ProgressBar1->Position++;
    //Se pintan los valores de las columnas
    if(!DBGrid1->DataSource->DataSet->FieldByName("Codi")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,1).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("Codi")->Value).c_str());
    if(!DBGrid1->DataSource->DataSet->FieldByName("Descripcio")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,2).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("Descripcio")-
>Value).c_str());
    if(!DBGrid1->DataSource->DataSet->FieldByName("Estoc")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,3).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("Estoc")-
>Value).c_str());
    if(!DBGrid1->DataSource->DataSet->FieldByName("Cost")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,4).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("Cost")->Value).c_str());
    if(!DBGrid1->DataSource->DataSet->FieldByName("PVP")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,5).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("PVP")->Value).c_str());
    if(!DBGrid1->DataSource->DataSet->FieldByName("Marca")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,6).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("Marca")-
>Value).c_str());
    if(!DBGrid1->DataSource->DataSet->FieldByName("Catalog")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,7).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("Catalog")-
>Value).c_str());

```

```

        if(!DBGrid1->DataSource->DataSet->FieldByName("Categoria")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,8).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("Categoria")-
>Value).c_str());
        if(!DBGrid1->DataSource->DataSet->FieldByName("IVA")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,9).OlePropertySet("Value",((AnsiStrin
g)DBGrid1->DataSource->DataSet->FieldByName("IVA")->Value).c_str());
        if(!DBGrid1->DataSource->DataSet->FieldByName("Marge")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,10).OlePropertySet("Value",((AnsiStri
ng)DBGrid1->DataSource->DataSet->FieldByName("Marge")-
>Value).c_str());
        if(!DBGrid1->DataSource->DataSet->FieldByName("DataAlta")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,11).OlePropertySet("Value",((AnsiStri
ng)DBGrid1->DataSource->DataSet->FieldByName("DataAlta")-
>Value).c_str());
        if(!DBGrid1->DataSource->DataSet->FieldByName("Imatge")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,12).OlePropertySet("Value",((AnsiStri
ng)DBGrid1->DataSource->DataSet->FieldByName("Imatge")-
>Value).c_str());
        if(!DBGrid1->DataSource->DataSet->FieldByName("Oferta")-
>Value.IsNull())
v1.OlePropertyGet("Item",InContador,13).OlePropertySet("Value",((AnsiStri
ng)DBGrid1->DataSource->DataSet->FieldByName("Oferta")-
>Value).c_str());

        InContador++;
        DBGrid1->DataSource->DataSet->Next();
    }

//this will stop asking you where you want to save your excel file or not
XL.OlePropertySet("DisplayAlerts",false);

```

```
//set the application as invisible, you can reverse "false" to "true" to
make it visible
XL.OlePropertySet("Visible",true);

//save what you have done
//XL.OleProcedure("Save");

//quite the excel application
//XL.OleProcedure("Quit");

//unassign variable
XL=Unassigned;
fGestioProductes->Panel3->Visible=false;
};
```

6. Prototip 2: clients

6.1. Requeriments

6.1.1. Client

- El sistema ha de permetre que l'actor pugui donar d'alta una nou client.
- El sistema ha de permetre que l'actor pugui editar les dades personals dels clients existents.
- El sistema ha de permetre ordenar les dades de dels clients disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa un client en cas que no tingui cap deute no hi hagi una pantalla de venda d'aquest client pendent de tancar.
- El sistema ha de permetre que l'actor pugui buscar un client del sistema.

- El sistema ha de permetre consultar el compte del client, per veure les vendes que té pendents de pagar.
- El sistema ha de comptabilitzar els punts de client i advertir de quan aquests són suficients per un regal de fidelització.

6.1.2. Venedor

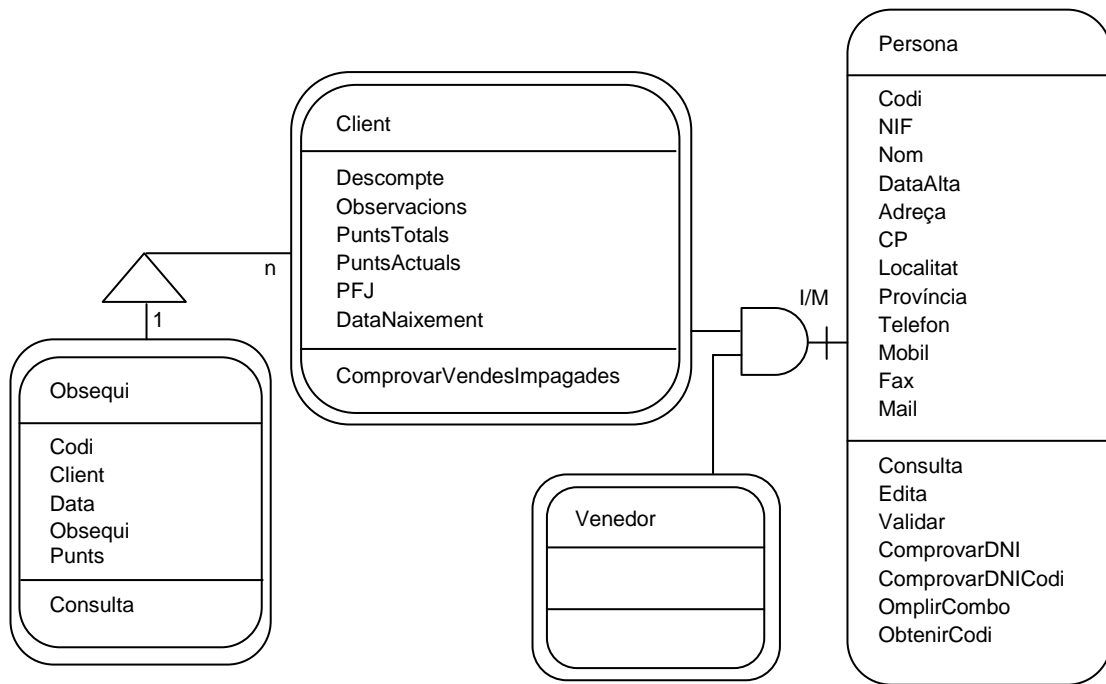
- El sistema ha de permetre que l'actor pugui donar d'alta un nou venedor.
- El sistema ha de permetre que l'actor pugui editar les dades personals dels venedors existents.
- El sistema ha de permetre ordenar les dades de dels venedors disponibles segons el camp especificat.
- El sistema ha de permetre que l'actor pugui donar de baixa un venedor en cas que no hi hagi cap pantalla de compra o de venda d'aquest venedor pendents de tancar.
- El sistema ha de permetre que l'actor pugui buscar un venedor del sistema.

6.1.3. Obsequi

- El sistema ha de permetre que l'actor pugui donar d'alta un nou obsequi.
- El sistema ha de permetre consultar els obsequis d'un client.

6.2. Anàlisi

A continuació mostrarem el diagrama de classes dels clients i els objectes analitzats en aquest apartat.



Persona

Codi: identificador del client; INTEGER.

NIF: NIF del client; VARCHAR.

Nom: Nom del client; INTEGER.

DataAlta: Data alta del client; INTEGER.

Adreça: Adreça del client; INTEGER.

CP: Codi Postal del client; INTEGER.

Localitat: Localitat del client; INTEGER.

Província: Província del client; INTEGER.

Telefon: Telefon del client; INTEGER.

Mobil: Mobil del client; INTEGER.

Fax: Fax del client; INTEGER.

Mail: Mail del client; INTEGER.

Consulta: Donada una taula, es mostra un llistat amb les dades de la taula de forma detallada ordenat per descripció.

Edita: Edita les dades d'un registre de la taula donat el seu codi.

Validar: Busca un registre de la taula donat el seu codi.

ComprovarDNI: Donada una taula i el NIF d'un registre de la taula, comprova si hi ha algun registre de la taula entrat a base de dades que tingui el mateix NIF.

ComprovarDNICodi: Donada una taula, el NIF i el codi d'un registre de la taula, comprova si hi ha algun registre de la taula amb el mateix NIF i codi diferent.

OmplirCombo: donada una taula, es mostren els seus registres amb un ComboBox.

ObtenirCodi: donada la descripció i la taula, retorna el codi del registre de la taula amb la descripció donada.

Client

DataNaixement: Data de naixement del client; INTEGER.

PFJ: persona física o jurídica; INTEGER.

Descompte: Descompte del client; INTEGER.

PuntsTotals: Punts totals dels clients; INTEGER.

PuntsActuals: Punts actuals dels clients; INTEGER.

Observacions: Observacions dels clients; INTEGER.

ComprovarVendesImpagades: Donat el codi del client comprova si té alguna venda pendent de pagar.

Obsequi

Codi: identificador de l'obsequi; INTEGER.

Client: el client al que pertany l'obsequi; VARCHAR.

Data: data quan s'entrega l'obsequi al client; INTEGER.

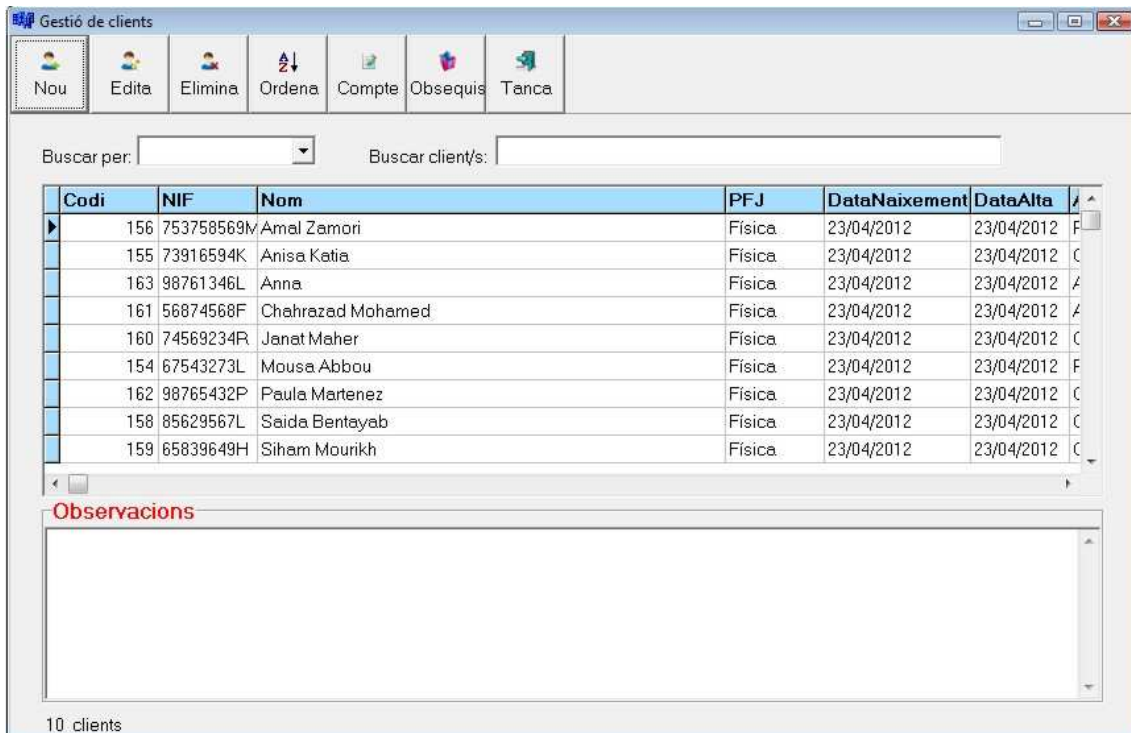
Obsequi: descripció del obsequi que s'entrega al client; INTEGER.

Punts: punts a descomptar dels punts actuals del client; INTEGER.

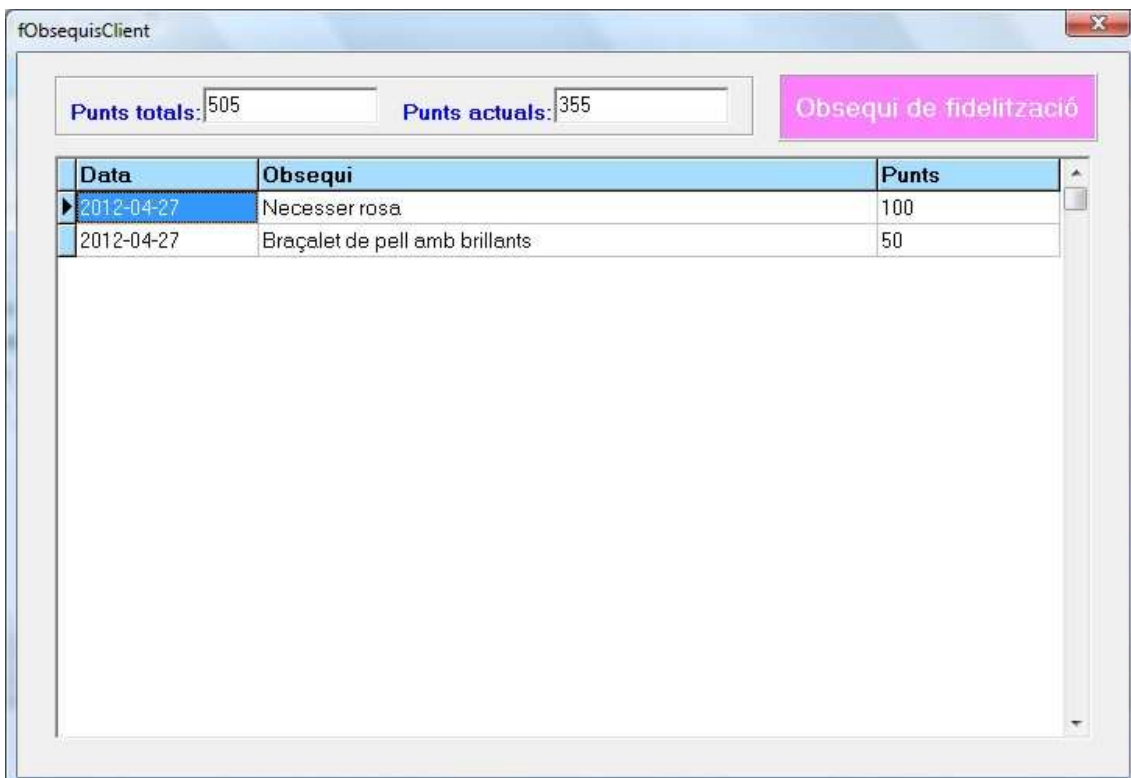
Consulta: Donat el codi del client, es mostra un llistat amb les dades dels obsequis del client de forma detallada.

6.3. Disseny del sistema

6.3.1. Interfícies



Pantalla de la gestió de clients.



Pantalla d'obsequis del client

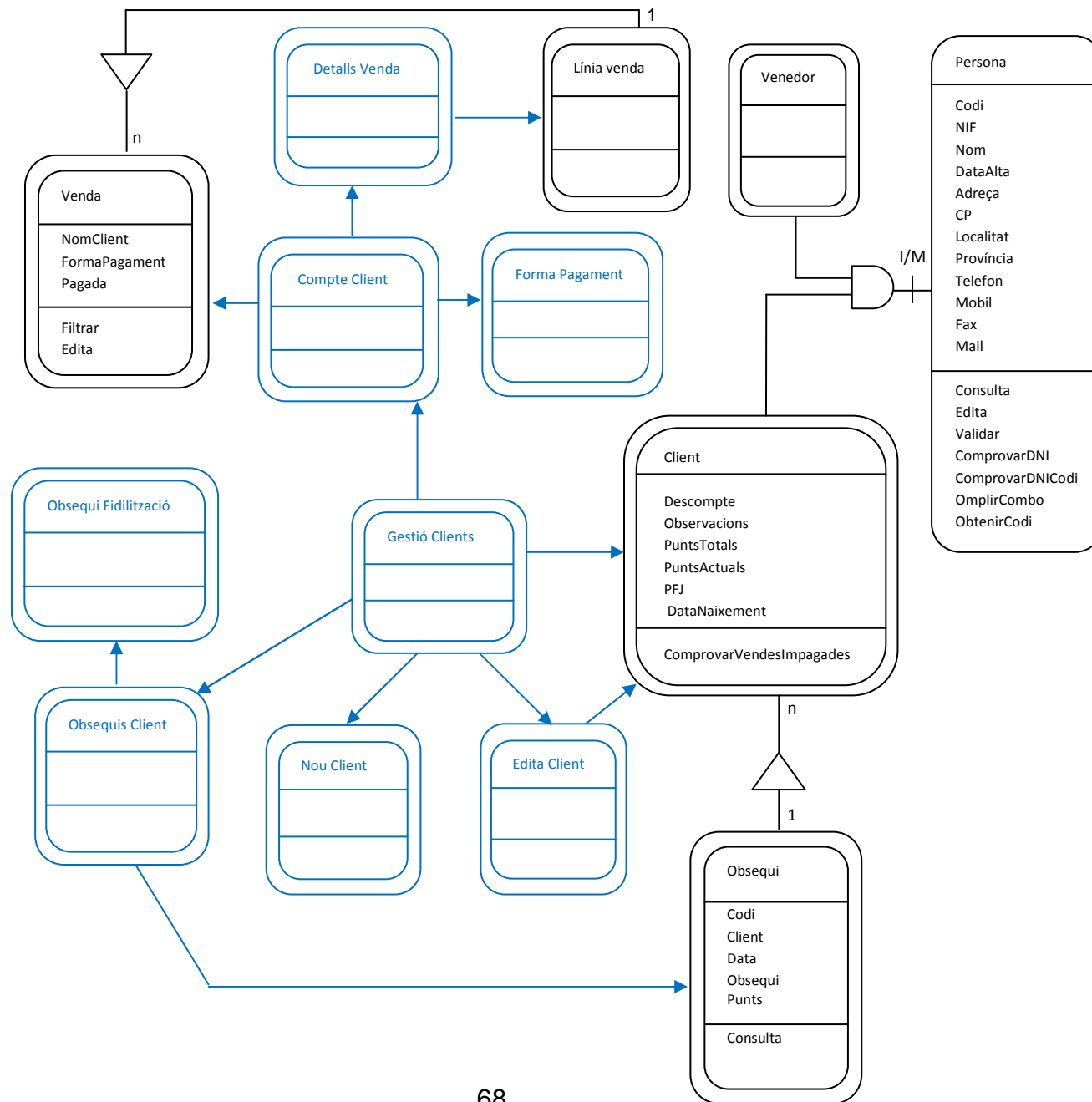
The screenshot shows a window titled 'Compte Client' with three buttons: 'Details', 'Paga', and 'Tanca'. Below the buttons is a table with the following data:

Codi	Data	Hora	Total	CodiVenedor	NomVenedor
4	26/04/2012	23:05:00	11,68	21	Safae Abbou El Hayani
5	26/04/2012	23:05:00	10,38	21	Safae Abbou El Hayani
6	26/04/2012	23:05:00	53,57	21	Safae Abbou El Hayani

At the bottom right of the window, a text box displays the total amount: 75.63 €.

Pantalla de compte del client

Seguidament, es mostrarà l'ampliació del diagrama de classe anterior.



6.3.2. Esdeveniments

NOM ESDEVENIMENT: Gestió de clients

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels clients.

DESCRIPCIÓ: es busquen les dades dels clients i es mostren per pantalla.

SERVEIS: Client.consulta.

NOM ESDEVENIMENT: Nou client

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades del nou client.

DESCRIPCIÓ: s'entren les dades del nou client i es guarda a la base de dades.

SERVEIS: Client.consulta, Client.alta.

NOM ESDEVENIMENT: Edita client

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les noves dades del client.

DESCRIPCIÓ: es modifiquen les dades del client i s'actualitzen a la base de dades.

SERVEIS: Client.consulta, Client.edita.

NOM ESDEVENIMENT: Compte client

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de les vendes del client pendents de pagar i el total a pagar.

DESCRIPCIÓ: es busquen les dades de les vendes que ha fet el client i que encara no s'han pagat, es calcula el total a pagar i es mostren per pantalla.

SERVEIS: Venda.consulta, Venda.sumarTotal.

NOM ESDEVENIMENT: Detalls venda

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran els detalls de la venda seleccionada.

DESCRIPCIÓ: es busquen els detalls de la venda seleccionada i es mostren per pantalla.

SERVEIS: LiniaVenda.consulta.

NOM ESDEVENIMENT: Forma de pagament

TIPUS ESDEVENIMENT: Informació

RESPOSTA: es modifica l'estat de la venda i la forma de pagament i es registren els canvis a la base de dades.

DESCRIPCIÓ: se selecciona la forma de pagament, es canvia l'estat de venda a pagada i es registren els canvis, després es mostren les vendes pendents de pagar de les que no s'han pagat si n'hi ha.

SERVEIS: Venda.edita, Venda.consulta.

NOM ESDEVENIMENT: Obsequis client

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels obsequis del client.

DESCRIPCIÓ: es busquen les dades dels obsequis del client i es mostren per pantalla.

SERVEIS: Obsequi.consulta.

NOM ESDEVENIMENT: Obsequi fidelització

TIPUS ESDEVENIMENT: informació

RESPOSTA: es registren les dades de l'obsequi del client.

DESCRIPCIÓ: s'entra l'obsequi del client, els punts que es descompten i la data que s'entrega l'obsequi, es resten els punts de l'obsequi dels punts actuals del client i es guarda l' informació a la base de dades.

SERVEIS: Obsequi.alta, Obsequi.consulta, Client.edita, Client.consulta.

6.4. Disseny de la base de dades

```
CREATE TABLE `client` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `NIF` varchar(10) character set latin1 NOT NULL,  
  `Nom` varchar(50) character set latin1 NOT NULL,  
  `PFJ` varchar(10) character set latin1 NOT NULL,  
  `DataNaixement` date NOT NULL,  
  `DataAlta` date NOT NULL,  
  `Adreca` varchar(50) character set latin1 NOT NULL,  
  `CP` varchar(10) character set latin1 NOT NULL,  
  `Localitat` varchar(50) character set latin1 NOT NULL,  
  `Provincia` varchar(50) character set latin1 NOT NULL,  
  `Telefon` varchar(15) character set latin1 NOT NULL,  
  `Mobil` varchar(15) character set latin1 NOT NULL,  
  `Fax` varchar(15) character set latin1 NOT NULL,
```

```
`Mail` varchar(50) character set latin1 NOT NULL,  
`Descompte` varchar(6) character set latin1 NOT NULL,  
`PuntsTotals` int(10) unsigned NOT NULL,  
`PuntsActuals` int(10) unsigned NOT NULL,  
`Observacions` varchar(1000) character set latin1 default NULL,  
PRIMARY KEY (`Codi`)  
)
```

```
CREATE TABLE `venedor` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `NIF` varchar(10) character set latin1 NOT NULL,  
  `Nom` varchar(50) character set latin1 NOT NULL,  
  `Adreca` varchar(50) character set latin1 NOT NULL,  
  `CP` varchar(10) character set latin1 NOT NULL,  
  `Localitat` varchar(50) character set latin1 NOT NULL,  
  `Provincia` varchar(50) character set latin1 NOT NULL,  
  `Telefon` varchar(15) character set latin1 NOT NULL,  
  `Mobil` varchar(15) character set latin1 NOT NULL,  
  `Fax` varchar(15) character set latin1 NOT NULL,  
  `DataAlta` date NOT NULL,  
  `Mail` varchar(50) character set latin1 NOT NULL,  
  PRIMARY KEY (`Codi`)  
)
```

```
CREATE TABLE `obsequi` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Punts` int(10) unsigned NOT NULL,  
  `Client` int(10) unsigned NOT NULL,  
  `Obsequi` varchar(100) NOT NULL,  
  `Data` date NOT NULL,  
  PRIMARY KEY (`Codi`)  
)
```


6.5. Disseny del programa

- **Exemple de sumar total**

```
void TfCompteClient::sumarTotal()
{
    fDataModule->cProveidorsProducte->Active=False;
    fDataModule->cProveidorsProducte->Close();
    fDataModule->cProveidorsProducte->SQL->Clear();
    fDataModule->cProveidorsProducte->SQL->Add("SELECT
Sum(venda.Total) AS SumaDeTotal FROM venda GROUP BY
venda.CodiClient,          venda.Pagada          HAVING
(((venda.CodiClient)="+fGestioClients->DBGrid1->DataSource->DataSet-
>FieldByName("Codi")->AsString+") AND ((venda.Pagada)='No'))");
    fDataModule->cProveidorsProducte->Prepare();
    fDataModule->cProveidorsProducte->Open();
    AnsiString          tot=fDataModule->cProveidorsProducte-
>FieldByName("SumaDeTotal")->AsString;
    fDataModule->cProveidorsProducte->Close();
    Edit1->Text=tot+" €";

};
```

7. Prototip 3: Vendes

7.1. Requeriments

7.1.1. Venda

- El sistema ha de permetre crear una venda, afegir-hi i treure qualssevol dels productes de la base de dades, així com assignar-hi un client i un venedor.
- El sistema ha de permetre veure l'històric de vendes.
- El sistema ha de permetre que l'actor pugui buscar les vendes fetes a un client del sistema entre dues dates.

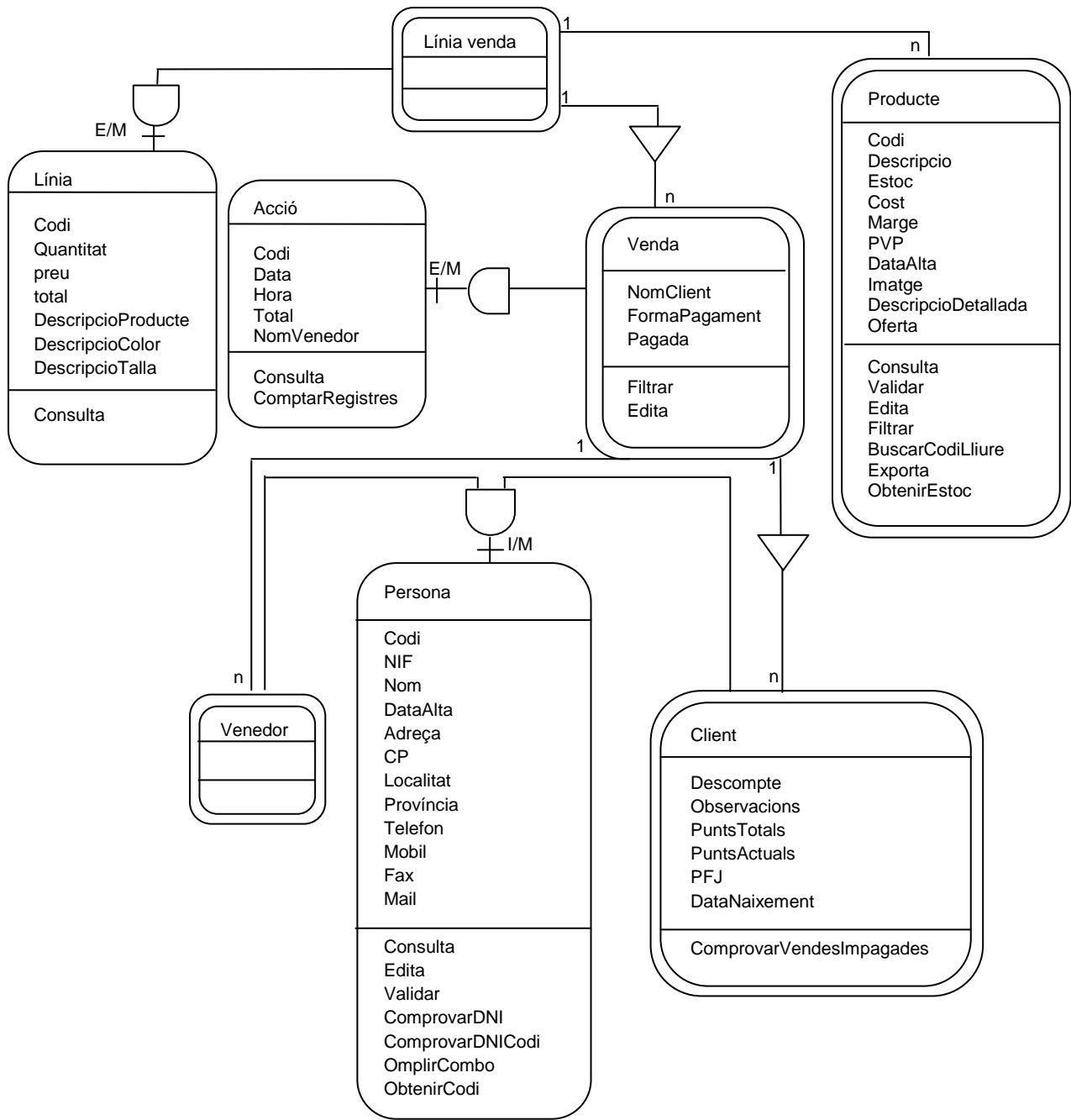
- El sistema ha de permetre filtrar les vendes per venedor/s entre dues dates.
- El sistema ha de permetre veure el llistat de clients deutors.
- El sistema ha de permetre veure els detalls d'una venda pendent de pagar.
- El sistema ha de permetre cobrar les vendes pendents de pagar.
- El sistema ha de permetre generar una factura d'una venda.

7.1.2. Línia de venda

- El sistema ha de permetre que l'actor pugui canviar el preu d'una línia de venda.
- El sistema ha de permetre que l'actor pugui canviar la quantitat d'una línia de venda.
- El sistema ha de permetre que l'actor pugui eliminar una línia de venda si encara no s'ha cobrat.
- El sistema ha de comprovar si el producte de la línia de venda té llista de talles i colors, en cas afirmatiu ha de permetre poder triar la talla i el color.
- El sistema ha de permetre mostrar l'estoc i l'imatge en cas que tingui d'un producte de la línia de venda.

7.2. Anàlisi

A continuació mostrarem el diagrama de classes de les vendes i els objectes analitzats en aquest apartat.



Acció

Codi: codi de l'acció; INTEGER.

Data: data que es fa l'acció; DATE.

Hora: hora que es fa l'acció; TIME.

Total: import total de l'acció; DOUBLE.

NomVenedor: nom del venedor que fa l'acció; VARCHAR.

Consulta: Donada una taula, es mostra un llistat amb les dades de la taula de forma detallada ordenat per data.

ComptarRegistres: Donada una taula, retorna la quantitat de registres que té.

Venda

NomClient: nom del client; VARCHAR.

FormaPagament: forma de pagament; VARCHAR.

Pagada: indica si s'ha pagat la venda o no; INTEGER.

Filtrar: Filtra les dades de vendes segons el client, el venedor i/o la data.

Edita: canvia l'estat de la venda a pagada.

Línia

Codi: codi de la línia; INTEGER.

Quantitat: quantitat de productes; INTEGER.

Preu: preu del producte; DOUBLE.

Total: preu total del producte; DOUBLE.

DescripcioProducte: descripció del producte; VARCHAR.

DescripcioColor: descripció del color; VARCHAR.

DescripcioTalla: descripció de la talla; VARCHAR.

Consulta: Donada una taula, es mostra un llistat amb les dades de la taula de forma detallada.

7.3. Disseny del sistema


7.3.1. Interfícies

Generar venda

Venedor: 19 Josep manel Gascon Data: 29/04/2012 Hora: 11:12

Client: 160 Janat Maher Adreça: C/ Sabdrella, 50, 2^a-2^a Desc: 30 % Punts: 214

Codi producte: Estoc actual: 1

Codi	Descripció	Quantitat	Preu	Color	Talla	Tc	Imatge
6	Sabata vestir pell peu punxegut	1	42,95	Negre	38	42	
4	Sandelles de tira creuades	1	47,79	Blanc	40	47	
5	Sandelles amb separador de pell	1	28,32	Plata	40	28	

Preu net: 119,06 €

Descompte: 35,72 €

Total: 83,34 €


✓ Acceptar ✗ Cancelar

Pantalla de generar venda

Seleccionar Producte

Buscar per: Buscar producte/s:

Codi	Descripció	Estoc	Cost	PVP	Marca	Categ	Categoria
55	Moneder Vintage Llaç	0	8,00	10,38	Blanco	Bosses i carteres	Moneders
53	Necesser amb tanca a pressió	4	5,00	6,37	Blanco	Bosses i carteres	Moneders
83	Penjoll	2	10,00	13,28	Bershka	Complements	Penjolls
93	Polsera	2	3,00	3,71	Promod	Complements	Polseras
59	polsera ampla	1	5,00	6,37	Promod	Complements	Polseras
92	Polsera de cadena	2	5,00	6,25	Bershka	Complements	Polseras
60	Polsera estil vintage	2	4,00	5,00	Tommy hilfiger	Complements	Polseras
68	Pulsera formas	2	6,00	7,65	Bershka	Complements	Polseras
91	Pulsera trenzada	3	4,00	5,00	Mango	Complements	Polseras
7	Sabata de vestir peep-toe de pell	3	30,00	49,91	Blanco	Calçat	Sabates di
8	Sabata de vestir punxeguda brillar	0	28,00	45,93	Lacoste	Calçat	Sabates di
18	Sabata Sala Peep Toe Llaç	4	40,00	70,80	Bershka	Calçat	Sabates di
16	Sabata Tacó Peep Toe Llaç	3	40,00	70,80	Lacoste	Calçat	Sabates di



✓ Acceptar ✗ Cancelar

Pantalla de seleccionar producte

Menú - [Crèdits]

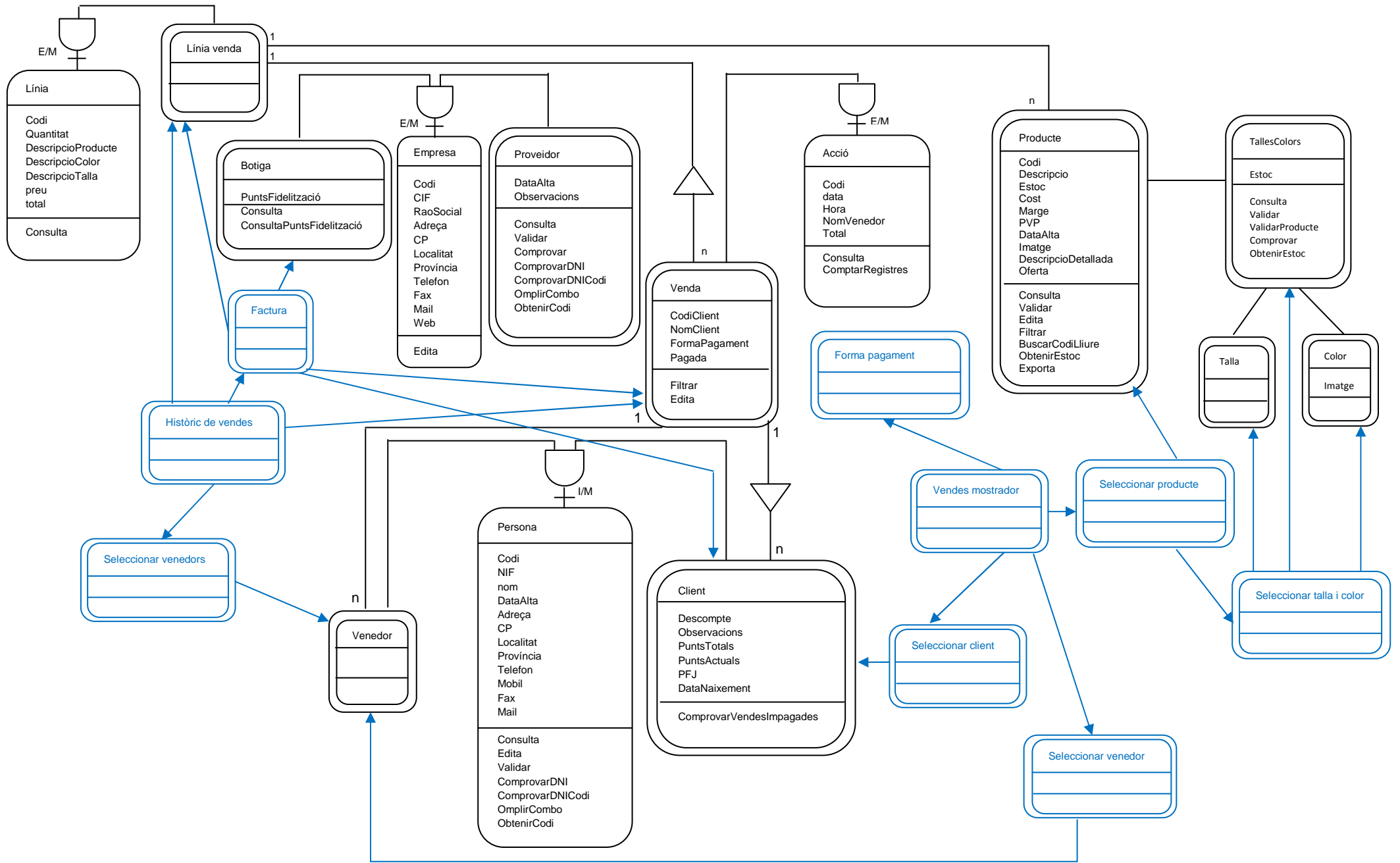
Filtres Moviments Llistats Caixa Finestra Ajuda Sortir

Codi client	Nom client	Adreça	Telefon	Total	Codi venedor	Nom venedor
163	Anna	Av. Sant Antoni, 32		7,65	21	Safae Abbou El Hayani
160	Janat Maher	C/ Sabdrella, 50, 2 ^a -2 ^a	953749834	37,27	19	Josep manel Gascon

diuenge, 29 de abril de 2012 20:14

Pantalla de crèdits

Seguidament, es mostrarà l'ampliació del diagrama de classe anterior.



7.3.2. Esdeveniments

NOM ESDEVENIMENT: Vendes mostrador

TIPUS ESDEVENIMENT: control

RESPOSTA: s'entren les dades de la venda i els seus detalls.

DESCRIPCIÓ: s'entra el venedor que fa la venda, el client i els productes que compra, de cada producte s'indica la quantitat la talla i el color si en té i es mostren per pantalla.

SERVEIS: calcularImports.

NOM ESDEVENIMENT: Seleccionar venedor

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels venedors i s'assignarà el venedor de la línia seleccionada a la pantalla de venda.

DESCRIPCIÓ: després de buscar les dades dels venedors i mostrar-les per pantalla, s'assigna el venedor seleccionat a la pantalla de venda.

SERVEIS: Venedor.consulta.

NOM ESDEVENIMENT: Seleccionar client

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels clients i s'assignarà el client de la línia seleccionada a la pantalla de venda.

DESCRIPCIÓ: després de buscar les dades dels clients i mostrar-les per pantalla, s'assigna el client seleccionat a la pantalla de venda.

SERVEIS: Client.consulta.

NOM ESDEVENIMENT: Seleccionar producte

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels productes i s'afegirà el producte de la línia seleccionada a la pantalla de venda.

DESCRIPCIÓ: després de buscar les dades dels productes i mostrar-les per pantalla, es comprova si el producte seleccionat té talles i colors, en cas afirmatiu es mostra un llistat de talles i colors del producte sinó s'afegirà el producte seleccionat a la pantalla de venda.

SERVEIS: Producte.consulta, Cataleg.consulta, Categoria.consulta, Marca.consulta, Iva.consulta.

NOM ESDEVENIMENT: Seleccionar talla i color

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les combinacions de talles i colors del producte i s'afegirà la combinació de talla i color de la línia seleccionada a la pantalla de venda.

DESCRIPCIÓ: després de buscar les diferents combinacions que hi ha de talles i colors del producte seleccionat i mostrar-les per pantalla, s'afegirà la combinació de talla i color de la línia seleccionada a la pantalla de venda.

SERVEIS: TallesColors.consulta, Talla.consulta, Color.consulta.

NOM ESDEVENIMENT: Factura

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de la factura.

DESCRIPCIÓ: es busquen les dades de la factura i es mostren en format PDF.

SERVEIS: Empresa.consulta, Client.consulta, Venda.consulta, LiniaVenda.consulta.

NOM ESDEVENIMENT: Historic de vendes

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de les vendes i els seus detalls entre dues dates.

DESCRIPCIÓ: es busquen les dades de les vendes entre dues dates a la part superior i els seus detalls a la part inferior i es mostren per pantalla, quan es canvia la posició del cursor sobre les vendes es mostren els detalls de la venda seleccionada, també es poden filtrar les vendes per dates, client i/o venedors.

SERVEIS: Venda.consulta, LiniaVenda.consulta, Venda.filtrar.

NOM ESDEVENIMENT: Seleccionar venedors

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels venedors.

DESCRIPCIÓ: es busquen les dades dels venedors i es mostren per pantalla.

SERVEIS: Venedor.consulta, Venda.filtrar.

7.4. Disseny de la base de dades

```
CREATE TABLE `venda` (  
  `Codi` int(10) unsigned NOT NULL,
```

```

`Data` date NOT NULL,
`CodiVenedor` int(10) unsigned NOT NULL,
`CodiClient` int(10) unsigned NOT NULL,
`Total` double NOT NULL,
`Hora` time NOT NULL,
`Pagada` varchar(2) NOT NULL,
`NomVenedor` varchar(45) NOT NULL,
`NomClient` varchar(45) NOT NULL,
`FormaPagament` varchar(20) NOT NULL,
PRIMARY KEY (`Codi`)
)

```

```

CREATE TABLE `liniavenda` (
`Codi` int(10) unsigned NOT NULL auto_increment,
`CodiProducte` int(10) unsigned NOT NULL,
`DescripcioProducte` varchar(45) NOT NULL,
`Quantitat` int(10) unsigned NOT NULL,
`Preu` double NOT NULL,
`Total` double NOT NULL,
`Venda` int(10) unsigned NOT NULL,
`CodiColor` int(10) unsigned NOT NULL,
`DescripcioColor` varchar(45) NOT NULL,
`CodiTalla` int(10) unsigned NOT NULL,
`DescripcioTalla` varchar(45) NOT NULL,
PRIMARY KEY (`Codi`)
)

```

7.5. Disseny del programa

- **Exemple de generar venda**

```

void __fastcall TfFormaPagament::AcceptarClick(TObject *Sender)
{

```

```

        if((fFormaPagament->Entrega->Text!="" | fFormaPagament->Targeta-
>Text!="" | fFormaPagament->Compte->Text!="") && fFormaPagament-
>Devolucio->Text.ToDouble()>=0){

            int nr=vend->comptarRegistres(fDataModule->Venda);

            //guardar venda

            vend=new Venda;

            vend->Codi=nr+1;

            vend->Data=fGenerarVenda->Data->Text;

            vend->Hora=fGenerarVenda->Hora->Text;

            vend->CodiVenedor=fGenerarVenda->CodiVenedor-
>Text.ToInt();

            vend->NomVenedor=fGenerarVenda->NomVenedor->Text;

            vend->CodiClient=fGenerarVenda->CodiClient->Text.ToInt();

            vend->NomClient=fGenerarVenda->NomClient->Text;

            vend->Total=fGenerarVenda->Total->Text.ToDouble();

            if(fFormaPagament->Compte->Text!=""){

                vend->FormaPagament="Compte";

                vend->Pagada="No";

            }else{

                //actualitzar caixa

                int c=cai->comptarRegistres();

                AnsiString con;

                if(fFormaPagament->Entrega->Text!=""){

                    vend->FormaPagament="Efectiu";

                    con="UPDATE Caixa SET TotalEfectiu=TotalEfectiu +
";

```

```

        }else if(
            fFormaPagament->Targeta->Text!=""){vend-
>FormaPagament="Targeta";
            con="UPDATE Caixa SET TotalTargeta=TotalTargeta+
";
        };
        vend->Pagada="Si";
        con=con+AnsiReplaceStr(fFormaPagament->Total-
>Text,",",",.");
        con=con+" WHERE Codi="+c;
        cai->edita(con);
    };
    vend->alta();

    for(int i=1;i<fGenerarVenda->StringGrid1->RowCount-1;i++){
        //guardar linies de venda
        lven=new LiniaVenda;
        lven->CodiProducte=fGenerarVenda->StringGrid1-
>Cells[0][i].ToInt();
        lven->DescripcioProducte=fGenerarVenda->StringGrid1-
>Cells[1][i];
        lven->Quantitat=fGenerarVenda->StringGrid1-
>Cells[2][i].ToInt();
        lven->Preu=fGenerarVenda->StringGrid1-
>Cells[3][i].ToDouble();
        lven->CodiColor=col->ObtenirCodi(fGenerarVenda-
>StringGrid1->Cells[4][i],fDataModule->Color);
    }
}

```

```

        lven->DescripcioColor=fGenerarVenda->StringGrid1-
>Cells[4][i];

        lven->CodiTalla=tal->ObtenirCodi(fGenerarVenda-
>StringGrid1->Cells[5][i],fDataModule->Talla);

        lven->DescripcioTalla=fGenerarVenda->StringGrid1-
>Cells[5][i];

        lven->Total=fGenerarVenda->StringGrid1-
>Cells[6][i].ToDouble();

        lven->Venda=nr+1;

        lven->alta();

        //actualitzar estoc de cada producte de la linia de venda

        if(pro->validar(fGenerarVenda->StringGrid1-
>Cells[0][i].ToInt())){

                fDataModule->Producte->Open();

                fDataModule->Producte->Edit();

                fDataModule->Producte->FieldByName("Estoc")-
>AsInteger=fDataModule->Producte->FieldByName("Estoc")->AsInteger-
fGenerarVenda->StringGrid1->Cells[2][i].ToInt();

                //actualitzar estoc de talla i color del producte si en té

                if(taco->validar(col->ObtenirCodi(fGenerarVenda-
>StringGrid1->Cells[4][i],fDataModule->Color),tal-
>ObtenirCodi(fGenerarVenda->StringGrid1->Cells[5][i],fDataModule-
>Talla),fGenerarVenda->StringGrid1->Cells[0][i].ToInt())){

                        fDataModule->TallesColors->Open();

                        fDataModule->TallesColors->Edit();

                        fDataModule->TallesColors-
>FieldByName("Estoc")->AsInteger=fDataModule->TallesColors-
>FieldByName("Estoc")->AsInteger-fGenerarVenda->StringGrid1-
>Cells[2][i].ToInt();

```

```

        fDataModule->TallesColors->Post();

        fDataModule->TallesColors->Close();

    };

    fDataModule->Producte->Post();

    fDataModule->Producte->Close();

};

//actualitzar punts client

if(cli->validar("Codi",fGenerarVenda->CodiClient-
>Text.ToInt(),fDataModule->Client)){

    fDataModule->Client->Open();

    fDataModule->Client->Edit();

    fDataModule->Client->FieldByName("PuntsTotals")-
>AsInteger=fDataModule->Client->FieldByName("PuntsTotals")-
>AsInteger+fGenerarVenda->Total->Text.ToDouble();

    fDataModule->Client->FieldByName("PuntsActuals")-
>AsInteger=fDataModule->Client->FieldByName("PuntsActuals")-
>AsInteger+fGenerarVenda->Total->Text.ToDouble();

    fDataModule->Client->Post();

    fDataModule->Client->Close();

};

};

Close();

fFormaPagament->netejarCampsEditables();

}else{

```

```
Application->MessageBoxA("L'import entrgat hauria de ser major  
o igual que el total!", "Atenció!", MB_OK+ MB_ICONERROR);  
  
};  
  
}
```

8. Prototip 4: Compres

8.1. Requeriments

8.1.1. Compra

- El sistema ha de permetre crear una compra, afegir-hi i treure qualssevol dels productes de la base de dades, així com assignar-hi un proveïdor i un venedor.
- El sistema ha de permetre veure l'històric de compres.
- El sistema ha de permetre que l'actor pugui buscar les compres fetes a un proveïdor entre dues dates.
- El sistema ha de permetre filtrar les compres per venedor/s entre dues dates.

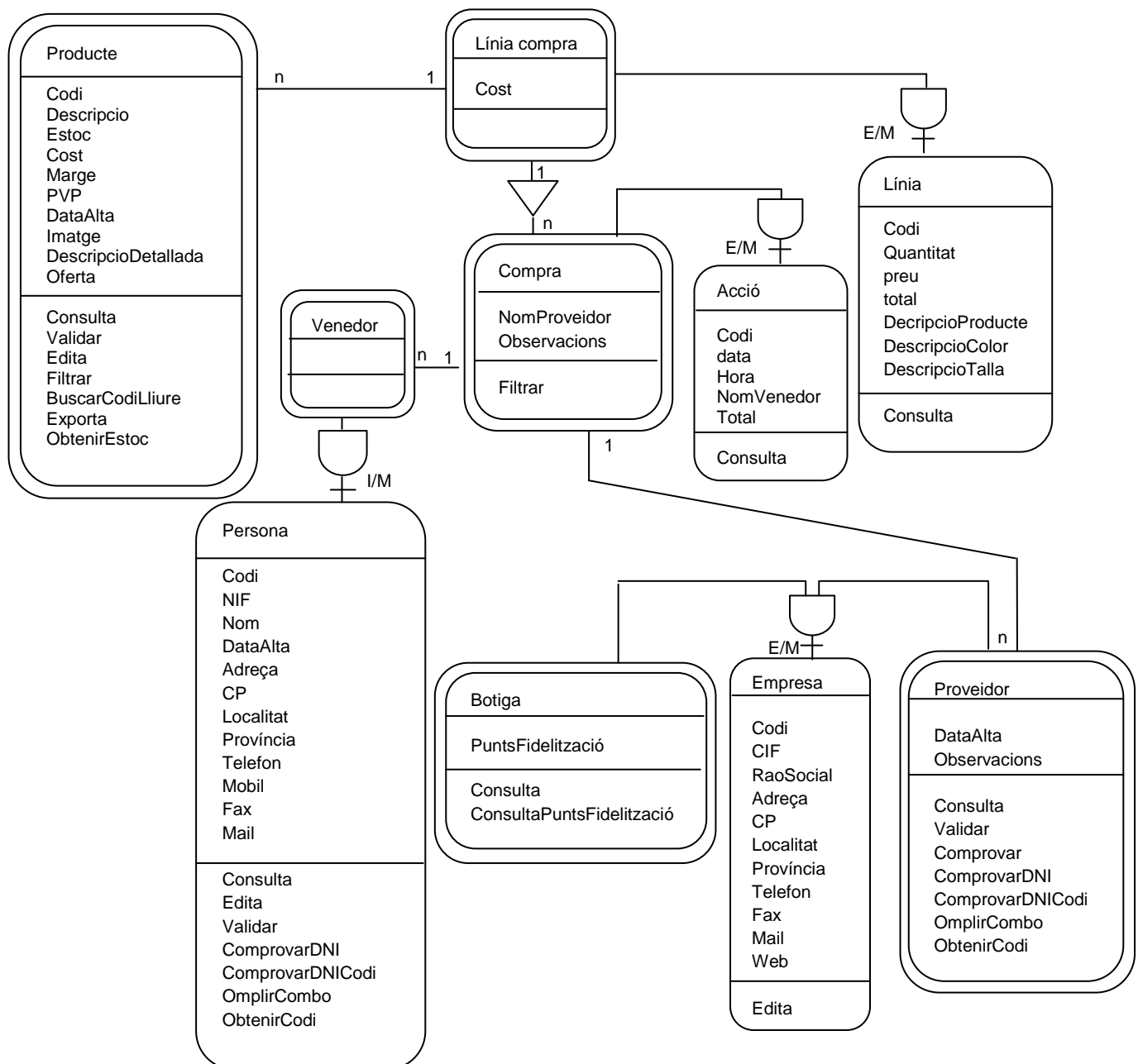
8.1.2. Línia de compra

- El sistema ha de permetre que l'actor pugui canviar els preus d'una línia de compra.
- El sistema ha de permetre que l'actor pugui canviar la quantitat d'una línia de compra.
- El sistema ha de permetre que l'actor pugui eliminar una línia de compra si encara no s'ha guardat.

- El sistema ha de comprovar si el producte de la línia de compra té llista de talles i colors, en cas afirmatiu ha de permetre poder triar la talla i el color.
- El sistema ha de permetre mostrar el marge de guany i l'imatge en cas que tingui d'un producte de la línia de compra.

8.2. Anàlisi

A continuació mostrarem el diagrama de classes de les compres i els objectes analitzats en aquest apartat.



Compra

NomProveïdor: nom del proveïdor; VARCHAR.

Observacions: observacions sobre la compra; VARCHAR.

Filtrar: Filtra les dades de comandes per venedor, Proveïdor/s, i/o dates.

Línia de compra

Cost: preu de cost del producte; DOUBLE.

Botiga

PuntsFidelització: màxim punts per a bescanviar per un obsequi; INTEGR.

Consulta: consulta les dades de la botiga.

ConsultaPuntsFidelitzacio: retorna els punts de fidelització.

Empresa

Codi: identificador de l'empresa; INTEGER.

CIF: CIF de l'empresa; VARCHAR.

RaoSocial: raó social de l'empresa; VARCHAR.

Adreca: adreça de l'empresa; VARCHAR.

CP: CP de l'empresa; VARCHAR.

Localitat: localitat de l'empresa; VARCHAR.

Província: Província de l'empresa; VARCHAR.

Telefon: telèfon de l'empresa; VARCHAR.

Fax: fax de l'empresa; VARCHAR.

Web: web de l'empresa; VARCHAR.

Mail: correu electrònic de l'empresa; VARCHAR.

Edita: Donada una taula edita les seves característiques.

8.3. Disseny del sistema

8.3.1. Interfícies

Menú - [Històric de compres]

Filters Moviments Llistats Caixa Finestra Ajuda Sortir

Data: Des de: 26/03/2012 Fins a: 29/04/2012 Proveïdor: Buscar per: Venedor: Tots Selecció

Codi	Data	Hora	Codi venedor	Nom venedor	Codi proveïdor	Nom proveïdor	Total
1	29/04/2012	17:41	20	Anna Martinez Lopez	46	Aymen Darkawi	70
2	29/04/2012	18:57	19	Josep manel Gascon	44	Dounia Mourikh El Hayani	111

Codi	Codi producte	Descripcio producte	Quantitat	Cost	Preu	Color	Talla	Total
89	79	Anell	5	7	9			35
90	81	Arrecades	10	4	5,1			40
91	72	Arrecades de gota	3	10	13,33			30
92	78	Arrecades vintage	1	6	7,65			6

diumenge, 29 de abril de 2012 18:58

Pantalla d'històric de compres

Seguidament, es mostrarà l'ampliació del diagrama de classe anterior.

8.3.2. Esdeveniments

NOM ESDEVENIMENT: Ingrés compres

TIPUS ESDEVENIMENT: control

RESPOSTA: s'entren les dades de la compra i els seus detalls.

DESCRIPCIÓ: s'entra el venedor que fa l'ingrés de compra, el proveïdor i els productes que se li compren, de cada producte s'indica la quantitat, els preus i la talla i el color si en té i es mostren per pantalla.

SERVEIS: calcularImports.

NOM ESDEVENIMENT: Seleccionar venedor

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels venedors i s'assignarà el venedor de la línia seleccionada a la pantalla de compra.

DESCRIPCIÓ: després de buscar les dades dels venedors i mostrar-les per pantalla, s'assigna el venedor seleccionat a la pantalla de compra.

SERVEIS: Venedor.consulta.

NOM ESDEVENIMENT: Seleccionar proveïdor

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels proveïdors i s'assignarà el proveïdor de la línia seleccionada a la pantalla de compra.

DESCRIPCIÓ: després de buscar les dades dels proveïdors i mostrar-les per pantalla, s'assigna el proveïdor seleccionat a la pantalla de compra.

SERVEIS: Proveïdor.consulta.

NOM ESDEVENIMENT: Seleccionar producte

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels productes i s'afegirà el producte de la línia seleccionada a la pantalla de compra.

DESCRIPCIÓ: després de buscar les dades dels productes i mostrar-les per pantalla, es comprova si el producte seleccionat té talles i colors, en cas afirmatiu es mostra un llistat de talles i colors del producte sinó s'afegirà el producte seleccionat a la pantalla de compra.

SERVEIS: Producte.consulta, Cataleg.consulta, Categoria.consulta, Marca.consulta, Iva.consulta.

NOM ESDEVENIMENT: Seleccionar talles i colors

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les combinacions de talles i colors del producte i s'afegirà la combinació de talla i color de la línia seleccionada a la pantalla de compra.

DESCRIPCIÓ: després de buscar les diferents combinacions que hi ha de talles i colors del producte seleccionat i mostrar-les per pantalla, s'afegirà la combinació de talla i color de la línia seleccionada a la pantalla de compra.

SERVEIS: TallesColors.consulta, Talla.consulta, Color.consulta.

NOM ESDEVENIMENT: Historic de compres

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades de les compres i els seus detalls entre dues dates.

DESCRIPCIÓ: es busquen les dades de les compres entre dues dates a la part superior i els seus detalls a la part inferior i es mostren per pantalla, quan es canvia la posició del cursor sobre les compres es mostren els detalls de la compra seleccionada, també es poden filtrar les compres per dates, client i/o venedors.

SERVEIS: Compra.consulta, LiniaCompra.consulta, Compra.filtrar.

NOM ESDEVENIMENT: Seleccionar venedors

TIPUS ESDEVENIMENT: control

RESPOSTA: es mostraran les dades dels venedors.

DESCRIPCIÓ: es busquen les dades dels venedors i es mostren per pantalla.

SERVEIS: Venedor.consulta, Venda.filtrar.

8.4. Disseny de la base de dades

```
CREATE TABLE `compra` (  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Data` date NOT NULL,  
  `CodiVenedor` int(10) unsigned NOT NULL,  
  `CodiProveidor` int(10) unsigned NOT NULL,  
  `Total` double NOT NULL,  
  `Hora` time NOT NULL,  
  `NomVenedor` varchar(45) NOT NULL,  
  `NomProveidor` varchar(45) NOT NULL,  
  `Observacions` varchar(1000) default NULL,  
  PRIMARY KEY (`Codi`)  
)
```

```

CREATE TABLE `liniacompra` (
  `Codi` int(10) unsigned NOT NULL auto_increment,
  `CodiProducte` int(10) unsigned NOT NULL,
  `DescripcioProducte` varchar(45) NOT NULL,
  `Quantitat` int(10) unsigned NOT NULL,
  `Cost` double NOT NULL,
  `Preu` double NOT NULL,
  `Total` double NOT NULL,
  `Compra` int(10) unsigned NOT NULL,
  `CodiColor` varchar(45) NOT NULL,
  `DescripcioColor` varchar(45) NOT NULL,
  `CodiTalla` varchar(45) NOT NULL,
  `DescripcioTalla` varchar(45) NOT NULL,
  PRIMARY KEY (`Codi`)
)

```

```

CREATE TABLE `botiga` (
  `Codi` int(10) unsigned NOT NULL auto_increment,
  `CIF` varchar(10) NOT NULL,
  `RaoSocial` varchar(45) NOT NULL,
  `Adreca` varchar(45) NOT NULL,
  `Localitat` varchar(45) NOT NULL,
  `Provincia` varchar(45) NOT NULL,
  `CP` varchar(10) NOT NULL,
  `Telefon` varchar(15) NOT NULL,
  `Fax` varchar(15) NOT NULL,
  `Mail` varchar(45) NOT NULL,
  `Web` varchar(45) NOT NULL,
  `Logo` varchar(45) NOT NULL,
  `PuntsFidelitzacio` int(10) unsigned NOT NULL,
  PRIMARY KEY (`Codi`)
)

```

8.5. Disseny del programa

- **Exemple de filtrar**

```
void Compra::filtrar(){
    fDataModule->Consulta->Active=False;
    fDataModule->Consulta->Close();
    fDataModule->Consulta->SQL->Clear();
    AnsiString d1=FormatDateTime( "YYYY-MM-DD", fHistoricCompres-
>desDe->Date) ;
    AnsiString d2=FormatDateTime( "YYYY-MM-DD", fHistoricCompres-
>finsA->Date) ;
    fDataModule->Consulta->SQL->Add("SELECT
Compra.Codi,Compra.Data,CAST(Compra.Hora AS CHAR(5))'Hora',
Compra.CodiVenedor 'Codi venedor',Compra.NomVenedor 'Nom venedor',
Compra.CodiProveidor 'Codi proveïdor',Compra.NomProveidor 'Nom
proveïdor', Compra.Total FROM Compra WHERE (Compra.Data
BETWEEN '"+d1+"' AND '"+d2+"'"));

    if(fHistoricCompres->buscarPer->Text=="Codi" && fHistoricCompres-
>Abuscar->Text!=""){
        fDataModule->Consulta->SQL->Add(" AND
Compra.CodiProveidor ='"+fHistoricCompres->Abuscar->Text);
    } else if(fHistoricCompres->buscarPer->Text=="Nom/Raó social" &&
fHistoricCompres->Abuscar->Text!=""){
        fDataModule->Consulta->SQL->Add(" AND
Compra.NomProveidor like '%" +fHistoricCompres->Abuscar->Text+"%'");
    };

    if(fSelVenedorsHisCom->ListBox2->Count>0){
        fDataModule->Consulta->SQL->Add(" AND
(Compra.NomVenedor='"+(AnsiString)fSelVenedorsHisCom->ListBox2-
>Items->Strings[0]+"''");
    };
    for(int i=1;i<fSelVenedorsHisCom->ListBox2->Count;i++){
```



```

        fDataModule->Consulta->SQL->Add(" OR
Compra.NomVenedor=""+(AnsiString)fSelVenedorsHisCom->ListBox2-
>Items->Strings[i]+""");
    };
    if(fSelVenedorsHisCom->ListBox2->Count>0){
        fDataModule->Consulta->SQL->Add("");
    };

    fDataModule->Consulta->SQL->Add(" Order By Compra.Data;");
    fDataModule->Consulta->Prepare();
    fDataModule->Consulta->Open();
    if(fDataModule->Consulta->RecordCount>0)lcom->consulta("SELECT
LiniaCompra.Codi, LiniaCompra.CodiProducte 'Codi producte',
LiniaCompra.DescripcioProducte 'Descripcio producte',
LiniaCompra.Quantitat,LiniaCompra.Cost, LiniaCompra.Preu,
CAST(LiniaCompra.DescripcioColor AS CHAR(25))'Color',
CAST(LiniaCompra.DescripcioTalla AS CHAR(25))'Talla',
LiniaCompra.Total FROM LiniaCompra WHERE
LiniaCompra.Compra="+fHistoricCompres->DBGrid1->DataSource-
>DataSet->FieldByName("Codi")->AsString);
        else lcom->consulta("SELECT LiniaCompra.Codi,
LiniaCompra.CodiProducte 'Codi producte',
LiniaCompra.DescripcioProducte 'Descripcio producte',
LiniaCompra.Quantitat,LiniaCompra.Cost, LiniaCompra.Preu,
CAST(LiniaCompra.DescripcioColor AS CHAR(25))'Color',
CAST(LiniaCompra.DescripcioTalla AS CHAR(25))'Talla',
LiniaCompra.Total FROM LiniaCompra WHERE LiniaCompra.Compra=0;");
    };

```

9. Prototip 5: Caixa

9.1. Requeriments

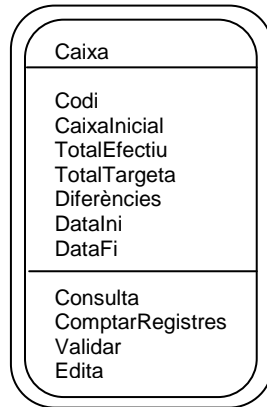
9.1.1. Caixa

- El sistema ha de permetre tancar la caixa del dia.

- El sistema ha de permetre veure l'històric de caixa.

9.2. Anàlisi

A continuació mostrarem el diagrama de classes de la caixa.



Caixa

Codi: codi de caixa; INTEGER.

DataIni: data inicial de la caixa; DATE.

DataFi: data final de la caixa; DATE.

CaixaInicial: import inicial que es deixa a la caixa; DOUBLE.

TotalEfectiu: total de vendes amb efectiu que hi ha a la caixa; DOUBLE.

TotalTargeta: total de vendes amb targeta que hi ha a la caixa; DOUBLE.

Diferències: el desquadre de caixa; DOUBLE.

Consulta: consulta les dades de la caixa pendent de tancar.

ComptarRegistres: compta la quantitat de registres de la caixa.

Validar: comprova que el codi de la caixa donada existeix.

Edita: Donada una taula, edita les dades de la taula.

9.3. Disseny del sistema

9.3.1. Interfícies

Caixa Diària

Caixa del dia

Dates

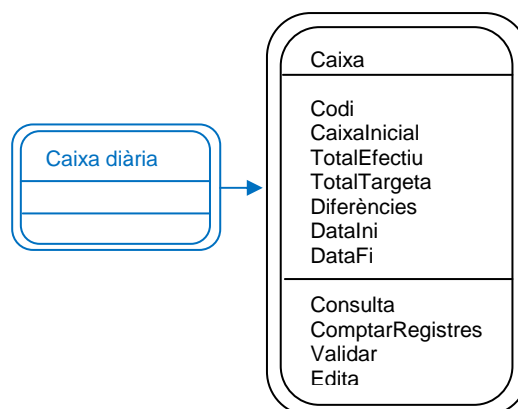
Des de: 29/04/2012 Fins a: 29/04/2012

Totals

Caixa Inicial:	280,45 €	Import d'obertura:	200,00 €
Total amb efectiu:	159,85 €	Import retirat:	240,00 €
Total amb targeta:	0,00 €		
Total caixa:	440,30 €	Diferències:	0,30 €

Pantalla de caixa diària

Seguidament, es mostrarà l'ampliació del diagrama de classe anterior.



9.3.2. Esdeveniment

NOM ESDEVENIMENT: Caixa diària

TIPUS ESDEVENIMENT: informació

RESPOSTA: es mostraran les dades de l'últim tancament de caixa.

DESCRIPCIÓ: es busquen les dades de l'últim tancament de caixa i es mostren per pantalla, s'entra l'import d'obertura i l'import retirat i es calculen les diferències.

SERVEIS: Caixa.edita, Caixa.alta.

9.4. Disseny de la base de dades

```
CREATE TABLE `caixa` (  
  `DataIni` date NOT NULL,  
  `DataFi` date NOT NULL,  
  `CaixaInicial` double unsigned NOT NULL,  
  `TotalEfectiu` double unsigned NOT NULL,  
  `TotalTargeta` double unsigned NOT NULL,  
  `Codi` int(10) unsigned NOT NULL auto_increment,  
  `Diferencies` double unsigned NOT NULL,  
  PRIMARY KEY USING BTREE (`Codi`)  
)
```

9.5. Disseny del programa

- **Exemple de comptarRegistres**

```
void __fastcall TfCaixaDiaria::AcceptarClick(TObject *Sender)  
{  
    if(fCaixaDiaria->importObertura->Text=="" | fCaixaDiaria-  
>importRetirat->Text=="")
```

```
Application->MessageBoxA("Atenció! L'import retirat i l'import  
d'obertura no es poden deixar en blanc.,"Advertència",MB_OK+  
MB_ICONWARNING);
```

```
else {  
  
    int c=cai->comptarRegistres();  
  
    AnsiString consulta;  
  
    consulta="UPDATE          Caixa          SET  
Diferencies="+AnsiReplaceStr(diferencies->Text,",",".")+"      WHERE  
Codi="+c+"";  
  
    cai->edita(consulta);  
  
    cai=new Caixa;  
  
    cai->DataIni=Now().CurrentDate();  
  
    cai->DataFi=Now().CurrentDate();  
  
    cai->CaixaInicial=fCaixaDiaria->importObertura-  
>Text.ToDouble();  
  
    cai->TotalEfectiu=0,00;  
  
    cai->TotalTargeta=0,00;  
  
    cai->Diferencies=0,00;  
  
    cai->alta();  
  
    Close();  
  
};  
  
}
```

10. Prototip 6: Llistats i resums estadístics

10.1. Sobre productes

Llistat de productes: es llisten les dades dels productes ordenades per descripció, es mostren el codi, la descripció, l'estoc i el preu.

Llistat de productes per marca: es llisten les dades dels productes agrupades per marca i dins de cada marca ordenades per descripció, de la marca es mostra la descripció i dels productes es mostren el codi, la descripció, l'estoc i el preu.

Llistat de productes per catàleg i categoria: es llisten les dades dels productes agrupades per catàleg i dins de cada catàleg agrupades per categoria i dins de cada categoria ordenades per descripció, del catàleg i de la categoria es mostren les descripcions i dels productes es mostren el codi, la descripció, l'estoc i el preu.

Llistat de productes amb estoc crític: es llisten les dades dels productes ordenades per descripció dels productes amb estoc crític, es mostren el codi, la descripció, l'estoc i el preu.

10.2. Sobre clients

Llistat de clients: es llisten les dades dels clients ordenades per descripció, es mostren el codi, el nom, l'adreça i el telèfon.

Llistat de vendes per client: es llisten les dades de les vendes del client seleccionat entre el rang de les dates seleccionades, les dades són ordenades pel codi. De la venda es mostra el codi, la data, el total, la forma de pagament i el detall de venda; en aquest últim es mostra el codi, la descripció, la quantitat, el preu i el total.

Llistat de clients amb deutes: es llisten les dades dels clients amb deutes ordenades per descripció, es mostren el codi, el NIF, el nom, l'adreça, el telèfon, i el total de deutes.

10.3. Sobre proveïdors

Llistat de proveïdors: es llisten les dades dels proveïdors ordenades per raó social, es mostren el codi, la raó social, l'adreça i el telèfon.

Llistat de productes a proveïdor: es llisten les dades dels productes comprats al proveïdor seleccionat entre el rang de les dates seleccionades ordenades per descripció, es mostren el codi, la descripció, la quantitat i el total.

10.4. Sobre vendes

Llistat de vendes: es llisten les vendes ordenades per dia i hora entre el rang de les dates seleccionades, es mostra la data, l'hora, el codi, el venedor, el client, i el total.

Llistat de vendes per marca: es mostra quantitat i el total de les línies de vendes fetes entre el rang de les dates seleccionades agrupades per marca.

Llistat de vendes per catàleg: es llisten les dades agrupades per línia de venda de les vendes fetes del catàleg seleccionat entre el rang de les dates seleccionades agrupades per categoria i de cada línia de venda es mostra el codi la descripció, la quantitat i el total.

Llistat de vendes per client: es llisten les dades de les vendes del client seleccionat entre el rang de les dates seleccionades, les dades són ordenades pel codi. De la venda es mostra el codi, la data, el total, la forma de pagament i el detall de venda; en aquest últim es mostra el codi, la descripció, la quantitat, el preu i el total.

Llistat de vendes per venedor per quantitat: es llisten les dades dels venedors ordenades per quantitat de vendes entre el rang de les dates seleccionades.

Llistat de vendes per venedor per total: es llisten les dades dels venedors ordenades per total de vendes entre el rang de les dates seleccionades.

Llistat de vendes per proveïdor: es llisten les dades dels proveïdors ordenades per quantitat de vendes entre el rang de les dates seleccionades.

10.5. Sobre compres

Llistat de compres: es llisten les dades de les compres fetes entre el rang de les dates seleccionades ordenades per data, es mostren la data, l'hora, el codi, el venedor que ha fet l'ingrés de la compra, el proveïdor al que s'ha comprat els productes, i el total de compra.

Llistat de compres per proveïdor: es llisten les dades dels productes comprats al proveïdor seleccionat entre el rang de les dates seleccionades agrupades per codi de compra i dins de cada compra ordenades per descripció del producte, de la compra es mostra el codi, la data i el total i dels productes es mostren el codi, la descripció, la quantitat i el total.

10.6. Sobre caixa

Llistat de caixa: es llisten les dades dels tancaments de caixa entre el rang de les dates seleccionades ordenades per data, es mostren la data del tancament de caixa, la caixa inicial, el total amb efectiu, el total amb targeta i les diferències.

10.7. Rànquing de vendes

Rànquing de vendes per producte per quantitat: el sistema crea un informe amb el rànquing dels 10 primers productes més venuts per quantitat durant per període indicat.

Rànquing de vendes per producte per preu: el sistema crea un informe amb el rànquing dels 10 primers productes més venuts per preu durant per període indicat.

Rànquing de vendes dels millors clients: el sistema crea un informe amb el rànquing dels 10 primers clients amb més volum de vendes generat durant el període indicat.

Rànquing de vendes per proveïdor: el sistema crea un llistat amb el rànquing dels 10 primers proveïdors en quant a unitats de vendes dels seus productes durant el període indicat.

11. Manual d'usuari

Per facilitar l'ús del programa, s'ha creat un manual d'usuari, que explica pas a pas totes les utilitats del sistema, el programa que s'ha fet servir per crear-lo és el HTML Help Workshop, és un programa molt senzill que genera l'arxiu d'ajuda en format .chm que consisteix en un índex, una taula de continguts i un conjunt de pàgines web HTML enllaçades a la taula.

Menú Productes

Quan s'activa la pantalla de la gestió de productes, es mostra un llistat ordenat per descripció de tots els productes que es disposen amb les seves dades, al canviar la posició del cursor en el llistat es mostra l'imatge i la descripció detallada del producte seleccionat cada vegada.

Gestió de productes

Nou Edita Elimina Ordena Exporta Tanca

Buscar per: Buscar producte/s:

Codi	Descripció	Estoc	Cost	PVP	Marca	Catàleg
1	Bota fins turmell	2	20,00	29,74	Lacoste	Calçat
2	Bota fins turmell	2	25,00	38,35	Tommy hilfiger	Calçat
3	Cuna carruse	1	20,00	29,50	Bershka	Calçat
4	Sandalies de tira creuades	2	30,00	47,78	Blanco	Calçat
5	Sandalies amb separador de pell	2	20,00	28,32	Mango	Calçat
6	Sabata vestir pell peu punxegut	2	28,00	42,95	Promod	Calçat
7	Sabata de vestir peep-toe de pell	3	30,00	49,91	Blanco	Calçat
8	Sabata de vestir punxeguda brillant	0	28,00	45,93	Lacoste	Calçat
9	Sabetes de vestir de pell amb el davant obert	2	30,00	51,33	Mango	Calçat

93 productes

Descripció detallada
Sabates de forma punxeguda brillant de vestir de color vermell, ideals per anar de festa.

Imatge

Pantalla del manual d'usuari

Part III
Especificació, anàlisi i disseny de
l'aplicació web

12. Descripció del projecte

Com a complement de la implantació del màrqueting, es vol crear una pàgina web per tal de fer promoció a la botiga ja que avui en dia és molt important existir en el món virtual.

Per aquesta pàgina web es volen implementar dues parts, una per l'administració del sistema, que estarà protegida amb una contrasenya i només serà accedida per l'administrador, en aquesta part s'ha de tenir l'opció de consultar, donar d'alta com editar les diferents taules de la base de dades del sistema, que són els catàlegs, les categories, les marques, les talles, els colors i els productes, dels productes hem de poder afegir o eliminar combinacions de talles i colors en cas que no constin, també hem de poder eliminar un producte en cas de no voler publicar-lo a la web o simplement perquè s'hagi acabat. També hem de tenir l'opció de consultar el llistat d'usuaris donats d'alta al sistema, i l'opció de visualitzar les comandes que ens han fet els usuaris, els detalls de cada comanda, el seu estat, així com poder canviar-ne l'estat.

I una altra part pública que tindrà accés per tothom que hi vulgui accedir, en aquesta part hem de poder veure els diferents catàlegs i categories dels que disposem a la botiga com els diferents productes que pertanyen a cada catàleg i categoria, també es podran fer encàrrecs a través de la pàgina web, però aquesta tasca serà reservada només pels usuaris donats d'alta al sistema, i per donar-se d'alta només s'ha d'entrar el nom i el correu electrònic, aquest últim es farà servir per comunicar a l'usuari l'estat de les seves comandes, totes les comandes que es fan a través d'Internet s'han d'anar a buscar a la botiga, un usuari també podrà veure l'historial de les seves comandes en qualsevol moment.

12.1. Requisits funcionals

Funcions de cerca: Dins de cada catàleg hi hauran les diferents categories que hi pertanyen i dins de les categories es mostraran tots els productes classificats. Qualsevol usuari que accedeixi a la pàgina web, estigui registrat o no, podrà buscar productes a través dels diferents catàlegs i categories

als que pertanyen o fent una cerca automatitzada, així com visualitzar els detalls de qualsevol producte. Tots els productes constaran d'una descripció, fotografia, preu i diferents combinacions de talles i colors si en tenen.

Funcions de comanda: Per poder fer comandes l'usuari s'haurà d'identificar i en cas que no estigui registrat haurà de realitzar l'operació de registre.

Funcions de gestió: Les Funcions de gestió estan reservades per l'administrador, aquest serà l'encarregat de realitzar les possibles altes, baixes i modificacions del sistema, a més podrà consultar el llistat de comandes, visualitzar els detalls de cada comanda i modificar el seu estat, així com llistar els productes, usuaris, catàlegs, categories, comandes...

12.2. Característiques de l'usuari

El nostre sistema constarà de tres tipus d'usuaris que són els que podran interactuar amb el sistema:

usuari públic: podrà realitzar consultes al catàleg de productes i navegar de forma lliure per la zona pública de la web. veure productes, descripcions de Productes ...

Usuari registrat: És aquell usuari que un cop identificat a més de les funcions esmentades anteriorment podrà realitzar comandes.

Usuari administrador: Aquest usuari serà l'encarregat de realitzar totes les funcions de gestió del sistema, modificacions, llistats, manteniment de la base de dades i comprovació del bon funcionament de la web....

13. Anàlisi

En aquest apartat i amb ajuda de la metodologia UML desenvoluparem un model de l'aplicació on diferenciarem als actors que interactuen amb els objectes del sistema mitjançant diferents relacions. Tot això ens ajuda a crear

un sistema robust i mantenible. La notació que s'utilitzarà és la proporcionada per l'estàndard UML. En el nostre cas usarem el diagrama de casos d'ús.

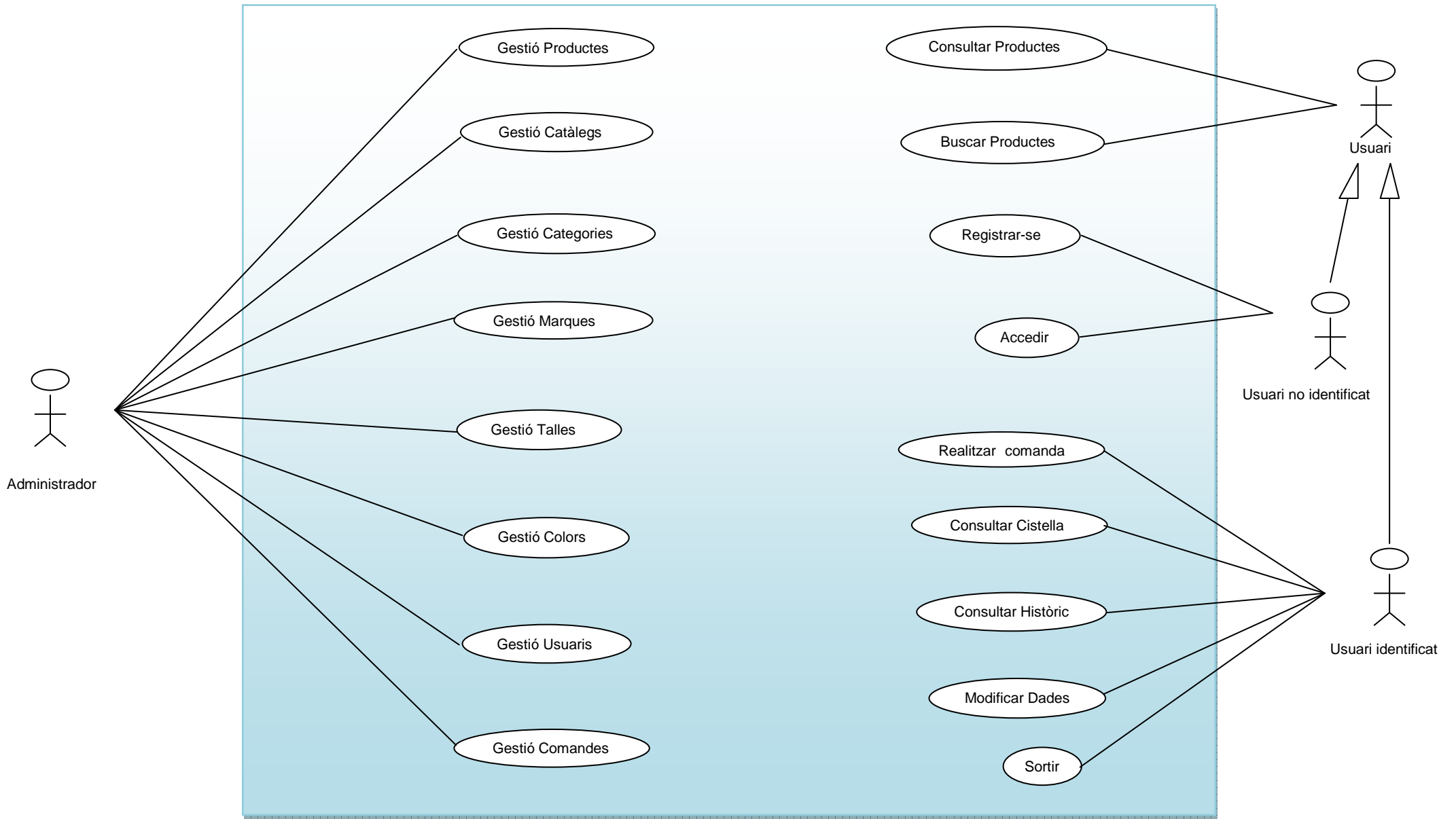
13.1. Diagrama de cassos d'ús

Un cas d'ús és una tècnica per a la captura de requisits potencials d'un nou sistema o una actualització de programari.

un cas d'ús és una seqüència d'interaccions que es desenvoluparan entre un sistema i els seus actors en resposta a un esdeveniment que inicia un actor principal sobre el propi sistema.

Els diagrames de casos d'ús serveixen per especificar la comunicació i el comportament d'un sistema mitjançant la seva interacció amb els usuaris i/o altres sistemes.

A continuació mostrarem el diagrama de casos d'ús de context i les fitxes de casos d'ús, només mostrarem algunes ja que hi ha que són molt semblants.



CAS D'ÚS:	Alta usuari
Versió	Visió usuari
Descripció	Un usuari es vol donar d'alta a la base de dades
Actors	Usuari no identificat
Precondició	No pot existir usuari amb el mateix email a la base de dades
Flux principal	<p>Triar l'opció de donar-se d'alta</p> <p>Omplir formulari amb: nom, email i contrasenya</p> <p>Pulsar acceptar</p> <p>Comprovar dades obligatòries formulari</p> <p>Comprovar format dades formulari</p> <p>Comprovar email no repetit</p> <p>Guardar dades usuari</p>
Flux alternatiu	Si falten dades obligatòries o el format de les dades entrades és incorrecte o l'email ja existeix retornar error
Postcondició	Si tot OK usuari donat d'alta altrament retornar error

CAS D'ÚS:	Modificar usuari
Versió	Visió usuari
Descripció	Un usuari vol modificar les seves dades
Actors	Usuari identificat
Precondició	No hi ha
Flux principal	<p>Triar l'opció de modificar dades</p> <p>Mostrar dades usuari</p> <p>Canviar dades formulari: nom, email i contrasenya</p> <p>Pulsar acceptar</p> <p>Comprovar dades obligatòries formulari</p> <p>Comprovar format dades formulari</p> <p>Modificar dades usuari</p>
Flux alternatiu	Si falten dades obligatòries o el format de les dades entrades és incorrecte retornar error
Postcondició	Si tot OK dades usuari modificades altrament retornar error

CAS D'ÚS:	Accedir usuari
Versió	Visió usuari
Descripció	Un usuari vol accedir al seu compte
Actors	Usuari no identificat
Precondició	Usuari existeix a la base de dades
Flux principal	<p>Triar l'opció d'accedir</p> <p>Omplir dades d'accés email i contrasenya</p> <p>Pulsar acceptar</p> <p>Comprovar dades obligatòries formulari</p> <p>Comprovar format dades formulari</p> <p>Comprovar usuari</p> <p>Accedir al compte d'usuari</p>
Flux alternatiu	Si falten dades obligatòries o el format de les dades entrades és incorrecte o usuari no existeix retornar error
Postcondició	Si tot OK usuari accedeix al seu compte altrament retornar error

CAS D'ÚS:	Afegir producte a la cistella
Versió	Visió usuari
Descripció	Un usuari vol afegir un producte a la seva cistella
Actors	Usuari identificat
Precondició	No hi ha
Flux principal	Triar la quantitat Triar la talla i el color si n'hi ha Pulsar comprar Comprovar existència Afegir producte a la cistella
Flux alternatiu	Si producte existeix augmentar quantitat
Postcondició	Si producte no existeix afegir a la cistella altrament augmentar quantitat

CAS D'ÚS:	Eliminar producte de la cistella de comandes
Versió	Visió usuari
Descripció	Un usuari vol eliminar un producte de la seva cistella
Actors	Usuari identificat
Precondició	El producte ha d'existir a la cistella
Flux principal	Pulsar eliminar Confirmar eliminació Eliminar producte de la cistella
Flux alternatiu	Si no es confirma eliminar no s'elimina producte de la cistella
Postcondició	Si tot OK s'elimina el producte de la cistella altrament no s'elimina

CAS D'ÚS:	Confirmar comanda
Versió	Visió usuari
Descripció	Es genera la comanda i es registra a la base de dades
Actors	Usuari identificat
Precondició	Com a mínim a d'existir un producte a la cistella
Flux principal	Pulsar confirmar Guardar dades comanda
Flux alternatiu	No hi ha
Postcondició	Es guarden les dades de la comanda amb estat pendent

CAS D'ÚS:	Veure producte
Versió	Visió usuari
Descripció	Un usuari vol consultar les dades d'un producte
Actors	Usuari, Usuari identificat, Usuari no identificat
Precondició	El producte ha d'existir a la base de dades
Flux principal	Seleccionar producte Buscar talles i colors del producte Mostrar dades producte
Flux alternatiu	Si el producte no té talles i colors informar de la incidència
Postcondició	Es mostren les dades del producte seleccionat

CAS D'ÚS:	Alta producte
Versió	Visió usuari
Descripció	L'administrador vol entrar un nou producte
Actors	Administrador
Precondició	El producte no ha d'existir a la base de dades i només pot pertanyer a un catalog i una categoria, i només pot tenir una marca i un tipus d'iva
Flux principal	Pulsar afegir producte mostrar catalegs, categories, marques i ives Omplir dades producte Pulsar acceptar Comprovar dades obligatòries formulari Comprovar format de les dades entrades Guardar producte
Flux alternatiu	Si falten dades obligatòries o el format de les dades entrades és incorrecte retornar error
Postcondició	Si tot OK Es guarden les dades del producte a la base de dades altrament retornar error

CAS D'ÚS:	Edita producte
Versió	Visió usuari
Descripció	L'administrador vol editar les dades d'un producte
Actors	Administrador
Precondició	El producte ha d'existir a la base de dades i només pot pertanyer a un catalog i una categoria, i només pot tenir una marca i un tipus d'iva
Flux principal	Pulsar editar producte Mostrar catalegs, categories, marques i ives Mostrar dades producte Modificar dades producte Pulsar acceptar Comprovar dades obligatòries formulari Comprovar format de les dades entrades Guardar dades producte
Flux alternatiu	Si falten dades obligatòries o el format de les dades entrades és incorrecte retornar error
Postcondició	Si tot OK Es guarden les dades del producte a la base de dades altrament retornar error

CAS D'ÚS:	Llistar comandes
Versió	Visió usuari
Descripció	L'administrador vol treure un llistat de comandes
Actors	Administrador
Precondició	No hi ha
Flux principal	Pulsar Llistat comandes Comprovar comandes Mostrar llistat comandes
Flux alternatiu	Si no hi ha comandes entrades a la base de dades informa de al incidència
Postcondició	Es mostra el llistat de comandes fetes pels usuaris

CAS D'ÚS:	Consultar detalls comanda
Versió	Visió usuari
Descripció	L'administrador vol consultar els detalls d'una comanda
Actors	Administrador
Precondició	Com a mínim ha d'existir una comanda
Flux principal	Seleccionar comanda Pulsar veure Mostrar detalls comanda
Flux alternatiu	No hi ha
Postcondició	Es mostren els detalls d'una comanda

14. Tecnologies

Per poder realitzar tota la nostra aplicació hem utilitzat diferents tecnologies i llenguatges de programació.

Inicialment hem necessitat simular un servidor web per poder utilitzar la nostra aplicació i veure les respostes que després es penjaran a un servidor real, dins de la xarxa.

Un servidor web o servidor HTTP és un programa informàtic que processa una aplicació del costat del servidor realitzant connexions bidireccionals i / o unidireccionals i síncrones o asíncrones amb el client generant o cedint una resposta en qualsevol llenguatge o Aplicació del costat del client. El codi rebut pel client sol ser compilat i executat per un navegador. Per a la transmissió de totes aquestes dades se sol utilitzar algun protocol. Generalment s'utilitza el protocol HTTP per a aquestes comunicacions, pertanyent a la capa d'aplicació del model OSI. El terme també s'empra per referir-se al ordinador que executa el programa.

Per això hem optat per instal·lar XAMPP, que incorpora en un sol programa:

- Apache, és el servidor web.
- PHP, llenguatge de programació interpretat, per a webs dinàmiques.
- MySQL, juntament amb PhpMyAdmin que ens permet la creació i gestió de la base de dades.

Per tal de penjar la pàgina web a Internet hem buscat un allotjament web, és a dir un servidor web on es pot publicar la pàgina web, i com a condició volíem que sigui un allotjament gratuït i que no hi hagi publicitat.

Els criteris que hem tingut en compte per escollir el hosting gratuït és fer una comparació entre aquestes característiques:

- Espai en disc
- Ample de banda
- Publicitat del hosting a les pàgines
- Possibilitat de posar publicitat pròpia

En la següent taula es resumeixen els hosting gratuïts analitzats pel que fa a les característiques que hem enumerat.

Hosting	Espai Disc	Ample Banda / mes	Publicitat hosting	publicitat pròpia
Host-ed	1000 MB	10 GB	No	Si
Zymic	6000 MB	50 GB	No	Si
000webhost	1500 MB	100 GB	No	Si

A partir de la comparació entre els diferents allotjaments web exposats anteriorment, es tria l'allotjament web 000webhost.

En el nostre disseny d'interfícies hem utilitzat la programació amb el llenguatge HTML i CSS, per donar-li una estil correcte a totes les pàgines i permetre amb el mínim treball poder fer modificacions.

JavaScript l'hem fet servir per obrir imatges i desar-les al servidor.

Per mostrar els colors d'una talla seleccionada d'un producte fem servir Ajax.

PHP és el llenguatge elegit per a desenvolupar la lògica de l'aplicació, ja que és un dels llenguatges més utilitzats a la xarxa, el qual ens permet la realització de pàgines dinàmiques de una forma ràpida i potent.

A al nivell de base de dades, hem utilitzat MySQL, que conjuntament amb PhpMyAdmin hem pogut crear i administrar la nostra base de dades d'una forma còmoda i visual.

14.1. HTML

Correspon a les sigles de (acrònim d'Hyper Text Markup Language, en català, "llenguatge de marcat d'hipertext"), és un llenguatge de marcat que deriva de l'SGML dissenyat per estructurar textos i relacionar-los en forma d'hipertext. Gràcies a Internet i als navegadors web, s'ha convertit en un dels formats més populars que existeixen per a la construcció de documents per a la web.

És usat per descriure l'estructura i el contingut en forma de text, així com per completar el text amb objectes tals com imatges, taules...

14.2. CSS

Cascading Style Sheets (CSS, en català Fulls d'Estil en Cascada) és un llenguatge de fulls d'estil utilitzat per descriure la semàntica de presentació (l'aspecte i format) d'un document escrit en un llenguatge de marques. La seva aplicació més comuna és dissenyar pàgines web escrites en HTML i XHTML,

CSS està dissenyat principalment per permetre la separació de contingut del document de la presentació del document, incloent-hi elements com la disposició, colors, i fonts. Aquesta separació pot millorar l'accessibilitat al contingut, proporcionar més flexibilitat i control en l'especificació de característiques de presentació, permetre que múltiples pàgines comparteixin un format comú, i redueix complexitat i repetició en el contingut estructural.

14.3. JavaScript

JavaScript és un llenguatge de programació interpretat. Es defineix com orientat a objectes. S'utilitza principalment en la seva forma al cantó del client, implementat com part d'un navegador web permetent millores en la interfície d'usuari i pàgines web dinàmiques.

JavaScript es dissenyat amb una sintaxi similar al llenguatge C, encara que adopta noms i convencions del llenguatge de programació Java.

Tots els navegadors moderns interpreten el codi JavaScript integrat dins de les pàgines web. Per interactuar amb una pàgina web es proveeix al llenguatge JavaScript d'una implementació del Document Object Model (DOM).

14.4. Ajax

AJAX són les sigles de Asynchronous Javascript And Xml, (JavaScript asíncron i XML), un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina. Això obre la porta a pàgines web interactives.

Habitualment les funcions d'Ajax es criden des del llenguatge JavaScript. Les dades són adquirides usant l'objecte XMLHttpRequest, que està disponible als llenguatges d'script que corren en navegadors moderns.

14.5. PHP

PHP és un llenguatge de programació interpretat que s'utilitza per a generar pàgines web de forma dinàmica. S'executa al cantó del servidor, tot i que és menys popular, també pot utilitzar-se per generar interfícies de comandes i aplicacions executables amb interfície gràfica.

Ens permet la construcció de pàgines web amb independència del servidor i de la base de dades que estiguem utilitzant, amb una compatibilitat completa en qualsevol plataforma, a més d'integrar-se de forma completa amb el codi HTML.

14.6. MySQL

MySQL, Structured Query Language, és un sistema de gestió de base de dades relacional, multi-fil i multiusuari. Una base de dades relacional emmagatzema les dades en taules separades, la qual cosa fa que la velocitat i flexibilitat de treball siguin molt grans.

MySQL ha esdevingut molt popular gràcies a la seva velocitat en executar consultes.

Es pot fer ús de MySQL en aplicacions de tota mena (web, d'escriptori o d'altres) de forma lliure i gratuïta sota les condicions de la llicència GPL.

15. Disseny i implementació

15.1. Base de dades

La pàgina web està desenvolupada amb el llenguatge PHP, de la base de dades del programa de gestió només es connecta a les taules relacionades amb la taula producte; que són catàleg, categoria, marca, talla i color, a part d'això es connecta a altres taules que només tenen relació amb l'aplicació web que són usuari, comanda i carrito.

```
CREATE TABLE `cataleg` (  
  
  `Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  
  `Descripcio` varchar(45) NOT NULL,  
  
  PRIMARY KEY (`Codi`)  
  
)
```

```
CREATE TABLE `categoria` (  
  
  `Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  
  `Descripcio` varchar(45) NOT NULL,  
  
  `Cataleg` int(10) unsigned NOT NULL,  
  
  `Imatge` varchar(250) DEFAULT NULL,  
  
  PRIMARY KEY (`Codi`)
```

)

CREATE TABLE `color` (

`Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,

`Descripcio` varchar(45) NOT NULL,

`Imatge` varchar(250) NOT NULL,

PRIMARY KEY (`Codi`)

)

CREATE TABLE `iva` (

`Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,

`Descripcio` varchar(45) CHARACTER SET latin1 NOT NULL,

`Tipus` varchar(5) CHARACTER SET latin1 NOT NULL,

PRIMARY KEY (`Codi`)

)

CREATE TABLE `marca` (

`Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,

`Descripcio` varchar(45) NOT NULL,

`Imatge` varchar(250) NOT NULL,

PRIMARY KEY (`Codi`)

)

CREATE TABLE `producte` (

```

`Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Descripcio` varchar(45) CHARACTER SET latin1 NOT NULL,
`Catalog` int(10) unsigned NOT NULL,
`Categoria` int(10) unsigned NOT NULL,
`Marca` int(10) unsigned NOT NULL,
`IVA` int(10) unsigned NOT NULL,
`DataAlta` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
`Imatge` varchar(250) CHARACTER SET latin1 DEFAULT NULL,
`DescripcioDetallada` varchar(1000) CHARACTER SET latin1 DEFAULT
NULL,
`Estoc` int(11) NOT NULL,
`Cost` double NOT NULL DEFAULT '0',
`Marge` double NOT NULL DEFAULT '0',
`PVP` double NOT NULL,
`Oferta` int(10) unsigned NOT NULL DEFAULT '0',
PRIMARY KEY (`Codi`)
)

```

```

CREATE TABLE `talla` (
`Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,
`Descripcio` varchar(45) NOT NULL,

```



```

PRIMARY KEY (`Codi`)
)

CREATE TABLE `tallescolors` (
  `Talla` int(10) unsigned NOT NULL,
  `Color` int(10) unsigned NOT NULL,
  `Estoc` int(11) NOT NULL,
  `Producte` int(10) unsigned NOT NULL,
  `Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`Codi`) USING BTREE
)

```

Les taules esmentades anteriorment són les mateixes que les de la base de dades del programa, i les que venen a continuació són exclusives per l'aplicació web; a la taula carrito guardem les línies de la cistella de comandes amb les dades del producte, a la taula comanda guardem les dades de la cistella de comandes, juntament amb les dades de l'usuari i la data, a la taula usuari guardem les dades dels usuaris que es donen d'alta al sistema, i a variable hi guardem l'iva.

```

CREATE TABLE `carrito` (
  `Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `Usuari` int(10) unsigned NOT NULL,
  `Producte` int(10) unsigned NOT NULL,
  `Quantitat` int(10) unsigned NOT NULL,
  `transaccioEfectuada` int(10) unsigned NOT NULL DEFAULT '0',

```

```

`Talla` int(10) unsigned NOT NULL,

`Color` int(10) unsigned NOT NULL,

PRIMARY KEY (`Codi`)

)

CREATE TABLE `comanda` (

`Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,

`Usuari` int(10) unsigned NOT NULL,

`Data` datetime NOT NULL,

`FormaPagament` int(10) unsigned NOT NULL,

`Total` double NOT NULL,

`Estat` int(10) unsigned NOT NULL DEFAULT '0',

PRIMARY KEY (`Codi`)

)

CREATE TABLE `usuari` (

`Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,

`Nom` varchar(50) CHARACTER SET latin1 NOT NULL,

`Email` varchar(100) CHARACTER SET latin1 NOT NULL,

`Estat` int(10) unsigned NOT NULL,

`Password` varchar(50) CHARACTER SET latin1 NOT NULL,

PRIMARY KEY (`Codi`)

```

)

```
CREATE TABLE `variable` (
```

```
  `Codi` int(10) unsigned NOT NULL AUTO_INCREMENT,
```

```
  `IVA` int(10) unsigned NOT NULL,
```

```
  PRIMARY KEY (`Codi`)
```

)

15.2. Detalls de la implementació

El programa utilitzat per la creació de la pàgina web és el l'Adobe Dreamweaver CS5.5, un programa que permet als usuaris disposar de vistes prèvies en múltiples navegadors, sempre que estiguin instal·lats a l'ordinador. També disposa d'algunes eines de gestió del lloc web, com la possibilitat de cercar i reemplaçar text o codi directament sobre tot el lloc web, així com l'ús i creació de plantilles per crear múltiples pàgines amb estructura similar. Disposa d'un Panell de comportaments que permet la utilització bàsica de JavaScript sense necessitat de coneixements de programació.

Una altra funcionalitat molt important són les eines de creació de contingut dinàmic en Dreamweaver. També permet la connexió a bases de dades com MySQL i Microsoft Access, per filtrar i mostrar el contingut utilitzant tecnologia de script.

L'aplicació web es compon per dues parts ben diferenciades, una part d'administració i una altra part pública:

La part d'administració només hi pot accedir l'administrador del sistema, i està protegida amb contrasenya, en aquesta part l'administrador disposa de la possibilitat de fer les diferents consultes, altes i modificacions del sistema, com per visualitzar els detalls de les comandes i canviar-ne l'estat.

A la part pública hi pot accedir tothom i és on es mostren els diferents productes que es disposen a la botiga segons el catàleg i la categoria als que pertanyen, també es permet donar-se d'alta al sistema, modificar les dades d'accés, fer comandes, consultar la cistella d'encàrrecs, i consultar l'historial de comandes (les tres últimes opcions estan reservades a usuaris registrats i identificats).

Com a mesura de seguretat per l'aplicació web es fa l'encriptació de la contrasenya quan l'usuari fa l'alta al sistema, la modificació de les seves dades o quan simplement vol accedir, l'algoritme que es fa servir és el Hash SHA-1. I pel que fa a la protecció de la part d'administració i la comprovació d'accés s'encarrega el mateix servidor web.

15.3. Exemples de la implementació

15.3.1. Accés

Des de acces.php realitzarem la comprovació que l'usuari existeix i que correspon amb la seva contrasenya, si alguna de les dades no són correctes ens redireccionarà a una altra pàgina on se'ns donarà la opció de tornar a introduir les dades. En el cas contrari la se'ns permetrà realitzar comandes, visualitzar la cistella i l'historial de comandes, i també se'ns donarà l'opció de modificar les nostres dades d'accés.

```
<?php
// *** Validate request to login to this site.
if (!isset($_SESSION)) {
    session_start();
}

$loginFormAction = $_SERVER['PHP_SELF'];
if (isset($_GET['accesscheck'])) {
    $_SESSION['PrevUrl'] = $_GET['accesscheck'];
}
}
```

```

if (isset($_POST['Email'])) {
    $loginUsername=$_POST['Email'];
    $password=$_POST['Password'];
    $pasEncriptada=sha1($password);
    $MM_fldUserAuthorization = "";
    $MM_redirectLoginSuccess = "acces_ok.php";
    $MM_redirectLoginFailed = "acces_error.php";
    $MM_redirecttoReferrer = false;
    mysql_select_db($database_conexion, $conexion);

    $LoginRS__query=sprintf("SELECT Codi, Email, Password FROM
usuari WHERE Email=%s AND Password=%s",
    GetSQLValueString($loginUsername, "text"),
    GetSQLValueString($pasEncriptada, "text"));

    $LoginRS = mysql_query($LoginRS__query, $conexion) or
die(mysql_error());
    $row_LoginRS = mysql_fetch_assoc($LoginRS);
    $loginFoundUser = mysql_num_rows($LoginRS);
    if ($loginFoundUser) {
        $loginStrGroup = "";

        if (PHP_VERSION >= 5.1) {session_regenerate_id(true);} else
{session_regenerate_id();}
        //declare two session variables and assign them
        $_SESSION['MM_Username'] = $loginUsername;
        $_SESSION['MM_UserGroup'] = $loginStrGroup;
        $_SESSION['MM_Codi'] = $row_LoginRS["Codi"];

        if (isset($_SESSION['PrevUrl']) && false) {
            $MM_redirectLoginSuccess = $_SESSION['PrevUrl'];
        }
        header("Location: " . $MM_redirectLoginSuccess );
    }
}

```

```

else {
    header("Location: ". $MM_redirectLoginFailed );
}
}
?>

```

15.3.2. Connexió

A l'arxiu `conexion.php` hi trobem les dades necessàries per establir la connexió amb el servidor, per mantenir les dades d'accés de l'usuari en totes les pàgines web sense anar posant-ho en cada pàgina per separat, lo que fem és incloure les dades de validació de la variable de sessió al principi de l'arxiu `conexion.php`, ja que aquest està inclòs en totes les pàgines `*.php`, i al final de l'arxiu `conexion.php` hi trobem un `include` de l'arxiu `funcions.php`, que és on tenim les funcions necessàries per fer anar l'aplicació, com poden ser les comprovació i obtenció de les diferents dades...

```

<?php if (!isset($_SESSION)) {

    session_start();

}??>

<?php

# FileName="Connection_php_mysql.htm"

# Type="MYSQL"

# HTTP="true"

$hostname_conexion = "localhost";

$databse_conexion = "basetfc";

$username_conexion = "root";

$password_conexion = "";

```

```

$conexion          =          mysql_pconnect($hostname_conexion,
$username_conexion,          $password_conexion)          or
trigger_error(mysql_error(),E_USER_ERROR);

?>

<?php

if(is_file("includes/funcions.php")){

    include("includes/funcions.php");

}else{

    include("../includes/funcions.php");

}

?>

```

15.3.3. Obrir imatges

```

<script>

function obrirImatge(){

    self.name = 'opener';

    remote = open('gestioimatge.php' ,'remote', 'width=400,
height=150,location=no,scrollbars=yes,menubars=no,tollbars=no,resizab
le=yes,fullscreen=no,status=yes');

    remote.focus();

}

</script>

```

Aquesta funció s'encarrega de definir el nom de la finestra, i la variable `remote` lo que fa és obrir la finestra amb les característiques que se li passen. En la finestra de **gestioimatge** trobem:

```

<?php if((isset($_POST["Enviat"])) && ($_POST["Enviat"]=="form1")){

    $nom_archiu = $_FILES['userfile']['name'];

    move_uploaded_file($_FILES['userfile']['tmp_name'], "../documents
/productes/" . $nom_archiu);?>

    <script>

        opener.document.form1.imatge.value="<?php echo $nom_archiu;
?>";

        self.close();

    </script>

    <?php}

else { ?>

<form action="gestioimatge.php" method="post" enctype="multipart/form-
data" id="form1">

    <p><input name="userfile" type="file" /></p>

    <p><input type="submit" name="button" id="button" value="Obrir
imatge" />

</p><input type="hidden" name="Enviat" value="form1" /></form>

<?php }?>

```

Per enviar dades de tipus binari com pot ser una imatge, un pdf... es fa amb `enctype="multipart/form-data"`, quan l'usuari selecciona una imatge i preta el botó obrir imatge, es guarda a la carpeta indicada i s'assigna el nom al camp imatge.

15.3.4. Enviament de correu electrònic

```
function enviamentCorreuHTML($destinatario, $contenido, $asunto)
```



```

{
    $mensaje='<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Documento sin título</title>

</head>

<body>

<table width="100%" border="0" cellspacing="3" cellpadding="3">

    <tr>

        <td></td>

    </tr>

    <tr>

        <td><p>Estimat Client:</p>

        <p>';

        $mensaje.=$contenido;

        $mensaje.='</p></td>

    </tr>

    <tr>

        <td>Moltes gràcies per la teva comanda a Boutique
Shahrazad, per qualsevol dubte pots contactar amb nosaltres
mitjançant el nostre correu electrònic:<br/> <a
href="mailto:d.ounia7@hotmail.com">d.ounia7@hotmail.com</a></td>

```

```

</tr>

</table>

</body>

</html>

';

    // Para enviar correo HTML, la cabecera Content-type debe
definirse

    $cabeceras = 'MIME-Version: 1.0' . "\n";

    $cabeceras .= 'Content-type: text/html; charset=iso-8859-1' . "\n";

    // Cabeceras adicionales

    $cabeceras .= 'From: d.ounia7@hotmail.com' . "\n";

    $cabeceras .= 'Bcc: d.ounia7@hotmail.com' . "\n";

    // Enviarlo

    mail($destinatario, $asunto, $mensaje, $cabeceras);

}

```

15.3.5. Mostrar colors de la talla seleccionada d'un producte

```

<script>

function requerir(){

    try{

        req=new XMLHttpRequest();

    }catch(err1){

        try{

```

```

        req=new ActiveXObject("Microsoft.XMLHTTP");
    }catch(err2){
        try{
            req=new ActiveXObject("Msxml2.XMLHTTP");
        }catch(err3){
            req= false;
        }
    }
}

return req;
}

var peticion=requerir();

function llamarAjaxGETcol(){
    valor=document.getElementById("FTalla").value;

    var url="color.php?valor="+valor+"&&recordID="+<?php echo
    $row_dadesProducte['Codi']; ?>;

    peticion.open("GET",url,true);

    peticion.onreadystatechange =respuestaAjaxcol;

    peticion.setRequestHeader('Content-Type','application/x-www-form-
    urlencoded');

    peticion.send(null);
}

function respuestaAjaxcol(){
    if(peticion.readyState==4){

```

```

        if(peticion.status==200){

            //alert(peticion.responseText);

            document.getElementById("col").innerHTML=peticion.responseText;

        }else{

            alert("ha ocurrido un error"+peticion.statusText);

        }

    }

}

</script>

```

! a color.php tenim la consulta:

```
<?php require_once('Connections/conexion.php'); ?>
```

```
<br/>Tria'n el color<br/>
```

Colors:

```
<?php
```

```
global $database_conexion, $conexion;
```

```
mysql_select_db($database_conexion, $conexion);
```

```
$valor=$_GET['valor'];
```

```
$recordID=$_GET['recordID'];
```

```

$e=sprintf("SELECT      color.Codi,      color.Descripcio      FROM
talla,color,producte,tallescors  WHERE  tallescors.Talla=talla.Codi
AND      tallescors.Color      =color.Codi      AND
tallescors.Producte=producte.Codi AND producte.Codi = ".$recordID."
AND talla.Codi = ".$valor." GROUP BY color.Descripcio");

```

```
$re=mysql_query($e, $conexion) or die(mysql_error());  
echo'<select name="FColor" id="FColor" style="width:150px">';  
while($f=mysql_fetch_array($re)){  
    echo'<option value="'.$f['Codi'].'">'.$f['Descripcion'].'</option>';  
    }  
echo'</select>';  
?>
```

Part IV

Conclusions i treball futur

16. Planificació final

		2010				2011				2012				
Activitat	Unitat temps: setmana	Des	Gen	Feb	Mar	Abr	Mai	Jun	Jul	Ago	Set	Mar	Abr	Mai
		1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4	1 2 3 4
Prototip 1	Definició i estudi	P P P P P P												
	Creació base dades	R R R R R R												
	Anàlisi			P P P										
Prototip 2	Disseny			R R R										
	Proves				P P P									
	Anàlisi					R R R R R								
Prototip 3	Disseny					P P								
	Proves						R							
	Anàlisi						P							
Prototip 4	Disseny						P							
	Proves							R						
	Anàlisi							P P						
Prototip 5	Disseny							R R						
	Proves							P P						
	Anàlisi								R					
Prototip 6	Disseny								P					
	Proves									R				
	Anàlisi									P				
Pàgina web	Disseny										R			
	Proves										P			
	Pàgina web											R P P P		
Conclusions												R R R R R R R R R R R R R R		
												P		

P: significa el temps previst en realitzar una activitat .
R: significa el temps real en realitzar una activitat .

Diagrama de Gantt per la planificació inicial

17. Conclusions i possibles millores

Aquest projecte ha estat de gran utilitat per enriquir el meu coneixement sobre el procés de desenvolupament d'un programari específic per una empresa. Els coneixements que he adquirit en la meua carrera sobre l'especificació, l'anàlisi i el disseny orientat a objectes m'ha ajudat molt de cara al desenvolupament d'aquesta aplicació.

Realitzar el projecte ha estat molt interessant perquè he pogut accedir a les necessitats reals de l'empresa he hagut d'adaptar el programa als requeriments sol·licitats. També he obtingut coneixements sobre com es gestiona una botiga de roba utilitzant recursos informàtics bàsics. Aquest programa estarà en període de proves pel propietari per fer una primera valoració, i veure si s'han d'afegir noves utilitats al sistema.

Cal destacar també que aquest projecte ha estat molt útil per poder complementar la meua formació, sobre temes que no es tracten a classe, i que són necessaris per l'experiència professional.

També remarcar que ha estat un cas real i, per tant, les dificultats sorgides són molt diferents a les que ens podem trobar en qualsevol altra pràctica realitzada durant la carrera. Per tant aquesta experiència ha resultat molt enriquidora i m'ha donat un punt de vista més global.

I pel que fa al desenvolupament de l'aplicació web, vaig treballar amb eines que desconeixia completament, dintre les quals el llenguatge PHP. cosa que m'ha fet dubtar si l'elecció d'aquest llenguatge havia estat l'apropiat, més que res pel meu desconeixement, però m'ha aportat molt a nivell formatiu.

A més PHP és una eina que en un curt espai de temps m'ha permès un maneig amb soltesa, fins i tot jo mateix m'he sorprès dels avenços que he aconseguit.

A part del llenguatge PHP, he practicat sobre el maneig d'eines, com Dreamweaver, i he après a fer servir els fulls d'estil CSS i HTML.

Com a possibles millores per l'aplicació podem destacar:

- **Codi de barres:** el lector de codi de barres és una eina molt útil per una empresa com una botiga de roba ja que simplifica molt el procés de vendes, compres i consultes permetent d'aquesta manera més agilitat i rapidesa alhora de realitzar qualsevol acció de les anomenades anteriorment, per no disposar del lector no s'han pogut fer proves de funcionament.
- **Selecció d'idioma i moneda:** Una altra possible millora de cara a una possible expansió de l'aplicació, seria la possibilitat d'elegir entre diferents idiomes i al mateix temps, dins de cada idioma, tenir la possibilitat d'escollir la moneda en cada cas.
- **Sessions d'usuari:** Un aspecte que no estava previst en la descripció, ja que en el cas concret de la nostre aplicació no hi havia necessitat de fer-ho, però que podria resultar de molta utilitat en un futur. Es tractaria d'afegir diferents nivells i protegir cada els usuaris amb contrasenyes, amb la possibilitat de poder canviar-la cada un determinat període de temps.

I pel tema de la pàgina web podem destacar:

- **TPV virtual:** estaria molt bé poder convertir la pàgina web en un TPV virtual per tal de fer les vendes a través d'Internet, d'aquesta manera hi haurà un notable increment en el procés de vendes, ja que el fet d'existir a Internet dóna més capacitat i obre les portes de cara a tenir cada vegada més clients.
- **Bloc:** incloure un bloc destinat als comentaris personals i opinions. És una bona manera de comunicació amb el client, així com una porta d'entrada a aquelles persones que no coneixien el negoci fins aquest moment.

Bibliografía

http://es.wikipedia.org/wiki/Servidor_web

<http://www.slideshare.net/klucho19/ventajas-y-desventajas-de-apache-y-iis>

<http://www.slideshare.net/programadorphp/cmo-elegir-un-servidore-web>

<http://web.iespana.es/services/hebergement/>

<http://www.tripod.lycos.es/>

<http://es.wikipedia.org/wiki/SGBD>

<http://sheyla88.blogspot.es/>

<http://es.wikipedia.org/wiki/C%2B%2B>

http://es.wikipedia.org/wiki/Visual_Basic

<http://es.wikipedia.org/wiki/Perl>

<http://es.wikipedia.org/wiki/Python>

<http://es.wikipedia.org/wiki/Javascript>

<http://es.wikipedia.org/wiki/Ruby>

http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada

<http://es.wikipedia.org/wiki/Ajax>

<http://es.wikipedia.org/wiki/Html>