

Dedicatòria

A tots els que em van preguntar,
Què? Com portes el projecte?

Agraïments

- A l'Escola Universitària Politècnica de Mataró, per ser una excel·lent casa d'estudis.
- Als meus pares i avis per donar-me l'oportunitat d'estudiar una carrera i donar-me suport contínuament.
- A Jordi Diu per animar-me i ajudar-me en tot moment.
- A tota la gent que no ha deixat de recolzar-me.
- Al professor Josep Roure Alcobé, pels seus consells, orientació i paciència com a ponent d'aquest projecte.

Resum

Tallers Sacreu S.C.C.L és una petita empresa de reparació de vehicles que amb el pas del temps ha vist com el sistema de gestió del seu negoci ha quedat obsolet. Actualment funciona amb fulls de càlcul i processadors de text, i existeix una manca important d'ordre. Per tant, necessita modernitzar i estructurar la gestió el taller. El següent projecte pretén assolir l'objectiu de millorar la gestió, oferir a aquesta empresa l'agilització de tots els seus processos i evitar la pèrdua de dades mitjançant l'elaboració d'una aplicació que permetrà controlar de forma centralitzada tots les funcions i processos del taller, referents a la facturació.

Resumen

Talleres Sacreu S.C.C.L es una pequeña empresa de reparación de vehículos que con el paso del tiempo ha visto como el sistema de gestión de su negocio ha quedado obsoleto. Actualmente funciona con hojas de cálculo y procesadores de texto, y existe una carencia importante de orden. Por lo tanto, necesita modernizar y estructurar la gestión el taller. El siguiente proyecto pretende lograr el objetivo de mejorar la gestión, ofrecer a esta empresa la agilización de todos los sus procesos y evitar la pérdida de datos mediante la elaboración de una aplicación que permitirá controlar de forma centralizada todos las funciones y procesos del taller, referentes a la facturación.

Abstract

Talleres Sacreu S.C.C.L is a small company of repair of vehicles that with the passage of time one has seen as the system of management of its business has remained obsolete. Nowadays it works with leaves of calculation and processors of text, and an important lack of order exists. Therefore, the management needs to modernize and to construct the workshop. The following project tries to achieve the lens to improve the management, to improve to this company the streamlining of all them its processes and to improve the loss

of information by means of the production of an application that will allow to control of centralized form all the functions and processes of the workshop, modal to the invoicing.

Índex general

Dedicatòria.....	II
Agraïments	IV
Resum	VI
Índex general	VIII
Índex figures	1
Índex taules.....	1
1. Introducció	1
1.1. Descripció del client	1
1.2. Funcionament del taller.....	2
2. Objectius.....	3
2.1. Objectiu general	3
2.2. Objectius específics	3
3. Pressupost.....	5
3.1. Planificació	5
3.2. Cost	6
3.2.1. Personal	6
3.2.2. Software	6
3.2.3. Hardware	7
3.2.4. Total	7
4. Anàlisi	9
4.1. Restriccions funcionals	9
4.1.1. Clients	9
4.1.2. Contactes	9

4.1.3.	Bancs.....	9
4.1.4.	Vehicles.....	9
4.1.5.	Proveïdors	9
4.1.6.	Factures Clients.....	10
4.1.7.	Factures Proveïdors.....	10
4.1.8.	Altres.....	10
4.2.	Requeriments no funcionals	10
4.3.	Diagrama de casos d'ús	12
4.4.	Especificació de casos d'ús	14
4.4.1.	Gestió Clients.....	14
4.4.2.	Gestió Proveïdors.....	19
4.4.3.	Gestió Contactes	24
4.4.4.	Gestió Vehicles	28
4.4.5.	Gestió Factures Proveïdors	32
4.4.6.	Gestió Factures Clients	34
4.4.7.	Gestió de bancs	37
4.4.8.	Gestió Ordres de Taller.....	40
4.4.9.	Gestió Tarifes.....	42
4.4.10.	Gestió Països.....	45
4.4.11.	Gestió Províncies	47
4.4.12.	Gestió Poblacions.....	49
4.4.13.	Gestió Codis Postals	51
4.4.14.	Gestió Operacions.....	53
4.4.15.	Gestió Material.....	55
4.5.	Diagrama de Classes del Domini.....	59
4.6.	Digrames de seqüència del sistema	62
4.6.1.	Cas d'ús "Alta Ordre Taller"	62

4.6.2.	Cas d'ús "Alta Factura Client"	63
5.	Disseny	65
5.1.	Patró arquitectònic	65
5.1.1.	Capa Presentació	66
5.1.2.	Capa Aplicació	67
5.1.3.	Capa Domini	68
5.2.	Patrons de Disseny	69
5.2.1.	Singleton	69
5.2.2.	Factoria.....	70
5.2.3.	Indirecció.....	70
5.2.4.	Controlador	71
5.3.	Base de dades	71
5.3.1.	Model conceptual	73
5.3.2.	Model relacional.....	75
5.3.3.	Regles de negoci	77
5.3.4.	Seqüències	78
5.4.	Diagrama de seqüència del cas d'ús "AltaOrdreTaller"	79
6.	Implementació.....	81
6.1.	Interfície gràfica.....	81
6.2.	Controladors.....	84
6.3.	Interfícies i Factoria de Registres.....	85
6.4.	Persistència	85
7.	Conclusions	89
	Annex II- PSP (Procés Software Personal)	95
	Annex III – Contingut del CD	105
	Bibliografia.....	107

Índex figures

Figura 1: Gràfic Costos	8
Figura 2: Diagrama de casos d'ús	12
Figura 3: Diagrama de Classes del Domini.....	59
Figura 4: Interfícies i Factoria Registres	61
Figura 5: Diagrama de seqüència dels sistema (altaOrdreTaller)	62
Figura 6: Diagrama de seqüència del sistema (altaFacturaClient).....	63
Figura 7: Esquema MVC 4 capes.....	65
Figura 8: Patró Singleton a la classe ConnexioBD	69
Figura 9: Patró Factoria a FactoriaRegistres	70
Figura 10: Model Conceptual de la Base de Dades	73
Figura 11: Model Relacional de la Base de Dades.....	75
Figura 12: Diagrama de seqüència AltaVehicle 1	79
Figura 13: Diagrama altaOrdreTaller 2	80
Figura 14: Finestra seleccióAlta.....	81
Figura 15: JClalendar de Toedter	83
Figura 16: Finestra IDClient	83
Figura 17: Finestra altaVehicle	84

Índex taules

Taula 1: Planificació de tasques	5
Taula 2: Cost del personal	6
Taula 3: Cost del software	6
Taula 4: Cost del hardware	7
Taula 5: Cost Total	7
Taula 6: Cuadre-Resum d'equivalències entre Mètrica v3 i el PSP	104
Taula 7: Cuadre-Resum d'equivalències entre UP i el PSP	104

1. Introducció

1.1. Descripció del client

Aquesta aplicació està dissenyada i desenvolupada seguint les necessitats d'un client real, Tallers Sacreu S.C.C.L.

Tallers Sacreu, S.C.C.L és una empresa dedicada des de fa més de 25 anys a la reparació general d'automòbils, especialment de vehicles industrials, (camions, grues, bobcats, etc ...). Empresa d'àmbit local i situada al cap de munt de la riera de l'Eixample d'Arenys de Munt, s'ha consolidat amb una cartera de clients fixes d'arreu de la comarca, tant empreses com particulars. Estem parlant de més de 250 clients que al llarg de l'any passen a fer les revisions i reparacions al seus vehicles. El volum de factures entre l'any passat i aquest ha disminuït a causa de la crisi econòmica. S'ha calculat que aquest any es pot arribar, entre 320 i 350 factures, uns 80.000 euros bruts a l'any.

Pel que fa als proveïdors, actualment treballen amb més d'una cinquantena de proveïdors habituals i altres d'espòradics, depenent de les promocions que els fan (ofertes, ràpels,...). Entre els proveïdors amb els que treballa s'hi troben proveïdors de serveis (identificats pel pla comptable de 1991, amb l'identificador 410), proveïdors ordinaris, és a dir proveïdors de mercaderies, matèries primeres etc, (identificats amb el 400) i finalment amb Administracions Públiques (identificades amb el 47).

La necessitat d'aquest software es deu a que el sistema d'informació del tallers s'ha quedat obsolet, és a dir, treballen a fulls de càlcul i editors de text. El problema d'aquest sistema de treball és la pèrdua d'informació i el poc control de gestió sobre clients, proveïdors i la facturació. L'objectiu d'aquesta aplicació és resoldre aquest problema de la gestió i sobretot solucionar el problema de la pèrdua d'informació. A més, permetrà agilitzar els processos relacionats amb la gestió de clients, proveïdors i factures.

1.2. Funcionament del taller

Fins a dia d'avui el funcionament del taller és com el de fa molts anys, un taller petit de poble i el tracte amb els clients es cordial però alhora molt personalitzat i de confiança . Els clients s'apropen al taller o en cas de no poder portar el vehicle se'ls hi va a buscar al seu domicili. No es fa cap mena de pressupost previ formal, sinó que senzillament es fa de paraula i aproximatiu.

Cada cop que entra un vehicle al taller se li crea una fitxa de reparació amb la data, el numero de reparació, marca i matricula del vehicle. Sovint si el mateix client porta a reparar un vehicle, mentre encara n'hi ha algun altre seu al taller, s'aprofita la mateixa fitxa de reparació. Quan es parla de fitxa de reparació, fa referència al que posteriorment es la factura de reparació del vehicle o vehicles.

Pel que fa a les factures de clients, el venciment és de 30 o 60 dies. El clients decideixen si volen que el pagament de les factures li siguin domiciliades al seu compte bancari o pel contrari poden pagar de la forma que els hi sigui més còmode.

El tracte amb els proveïdors és excel·lent, treballen amb proveïdors propers al taller o que els hi poden servir els materials en qüestió d'hores. Aquest tracte amb els proveïdors és molt convenient que sempre resti excel·lent, ja que al ser un taller multi marca, és a dir, reparen vehicles de tot tipus i marques, i cada tipus i model de marca utilitza un material o peces diferents. No disposen d'estoc de material al taller.

2. Objectius

2.1. Objectiu general

Desenvolupar una aplicació:

- Escalable.
- Usable (Facilitat d'ús).
- Econòmica .
- A mida, pensada exclusivament per al sector dels petits tallers mecànics i més concretament en client,Tallers Sacreu, S.c.c.l, que no tenen un gran volum de treball ni guarden un estoc concret de cada material.
- Mantenible.
- Robusta.

2.2. Objectius específics

- Resoldre el problema de la gestió de clients, proveïdors i facturació.
 - Resoldre el problema de la pèrdua de dades.
 - Desenvolupar una interfície gràfica elaborada, per tal, de reduir al màxim els errors d'introducció de dades per part de l'usuari.
 - Conèixer internament la gestió d'un taller mecànic d'automòbils.
 - Desenvolupar una aplicació robusta, eficient i escalable.
 - Experimentar en primera persona el desenvolupament integral d'una aplicació per a un client real.
 - Conservar els principis bàsics de la programació orientada a l'objecte.
 - Ampliar coneixements en la plataforma J2SE.
 - Aplicar els coneixements adquirits al llarg de la carrera, en el desenvolupament d'un software de gestió, com la utilització de patrons arquitectònics, patrons de disseny, etc..
-

3. Pressupost

3.1. Planificació

Tasques	Duració (en hores)
Entrevistes amb el client	10 h
Especificació de requeriments	5 h
Diagrama de casos d'ús	1 h
Especificació de casos d'ús	10 h
Anàlisi d'eines i tecnologies	3 h
Anàlisi de l'estructura de l'aplicació	5 h
Elaboració pressupost	2 h
Diagrama de classes	15 h
Model conceptual	35 h
Programació de la interfície gràfica	50 h
Programació dels casos d'ús	240 h
Documentació	20 h
Testing	15 h
TOTAL	411 h

Taula 1. Planificació de tasques

3.2. Cost

3.2.1. Personal

Hores	Preu/hora
411 h	12 €/h
TOTAL	4.939 €

Taula 2. Cost del personal

3.2.2. Software

Producte	Preu
Microsoft Windows Vista Premium OEM	Inclòs en hardware
Microsoft Office Student 2007	88 €
NetBeansIDE 6.5.1	Gratuït
Sybase PowerDesigner 15	Versió Prova
MySQL	Gratuït
SQLyog	Gratuït
TOTAL	88 €

Taula 3. Cost del software

3.2.3. Hardware

Equip	Preu	Amortització (4 mesos)
Asus M51VA	850 €	283 €
TOTAL		283 €

Taula 4. Cost del hardware

3.2.4. Total

Concepte	SubTotal
Cost del personal	4.939 €
Cost del software	88 €
Cost del hardware	283 €
TOTAL	5.310 €

Taula 5. Cost Total

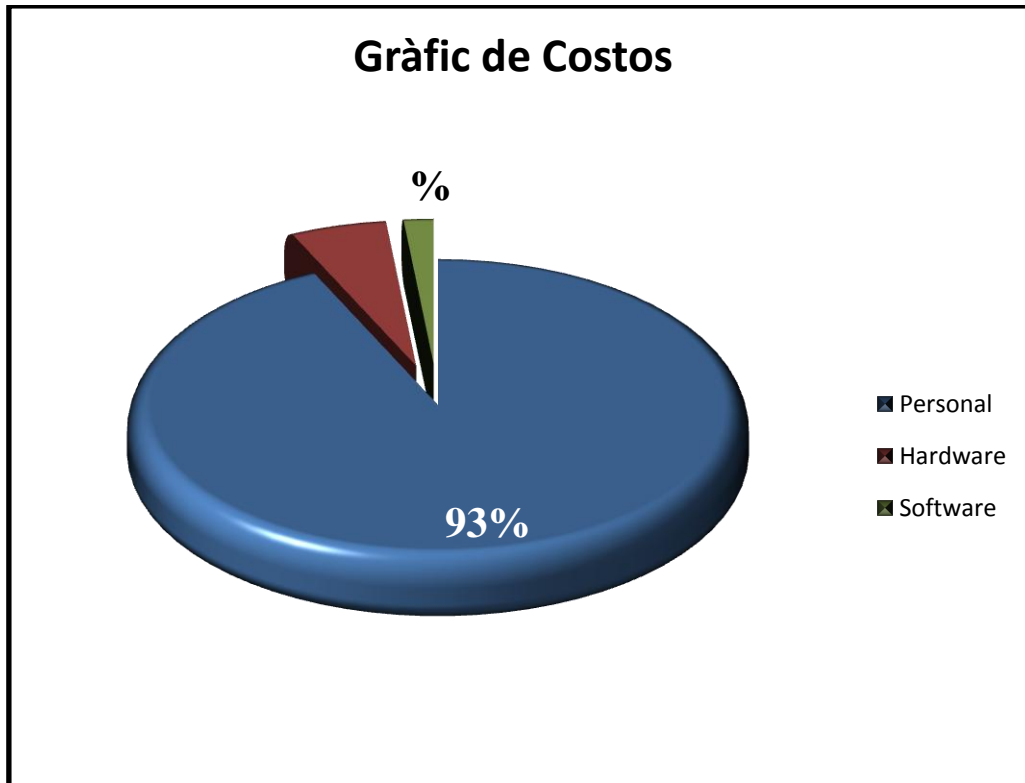


Figura 1. Gràfic Costos

4. Anàlisi

4.1. Restriccions funcionals

4.1.1. Clients

- L'identificador ha de començar per "43" i ha de tenir 6 dígits.
- No es poden donar de baixa clients amb factures relacionades, amb la diferència de la data actual a la data d'emissió de la factura menor a 5 anys.
- Si un client canvia de nif no es pot modificar, s'ha de crear un de nou.

4.1.2. Contactes

- No pot existir un contacte sense dades de telèfons, mòbils, faxes o correus electrònics.
- No pot existir un contacte sense client o contacte.
- No es pot crear un contacte d'un client o proveïdor donat de baixa.

4.1.3. Bancs

- Un client o proveïdor no poden tenir 2 bancs predefinits alhora.
- No es pot associar o crear un banc a un client o proveïdor donat de baixa.

4.1.4. Vehicles

- No pot existir un vehicle sense client.
- No es pot crear o modificar un vehicle a un client donat de baixa.

4.1.5. Proveïdors

- L'identificador ha de començar per "40", "41" o "47".
 - No es poden donar de baixa proveïdors amb factures relacionades, amb la diferència de la data actual a la data d'emissió de la factura menor a 5 anys.
 - Si un proveïdor canvia de nif no es pot modificar, s'ha de crear un de nou.
-

4.1.6. Factures Clients

- No es poden fer factures a cotxes donats de baixa.
- El venciment no pot ser anterior a la data d'emissió.
- Les factures sempre han de ser amb iva inclòs.
- El total d'una factura pot ser negatiu, per si alguna retroacció.
- Si hi ha ja hi ha factures introduïdes, el mes d'emissió de la següent factura ha de ser igual o posterior a les introduïdes al sistema.
- No es poden modificar si el llistat d'ives on esta inclòs s'ha tancat.

4.1.7. Factures Proveïdors

- No es poden crear factures de proveïdors donats de baixa.
- Els números de factura (identificador) han de correlatius al llarg de l'any i cada 1 de gener es torna a començar des de 1.
- El total de la factura pot ser negatiu si es un abonament.
- En el cas que la factura emesa per un proveïdor tingui diferents tants per cent, es crearà una nova factura proveïdor per a cada tant per cent.
- Una factura pot tenir iva i irpf alhora.

4.1.8. Altres

- Un codi postal només pot estar associat a una població.
- Una població només pot ser d'una província.
- Una província només és d'un país.
- No poden existir 2 tipus d'operacions iguals.
- No poden existir 2 tipus de material iguals.

4.2. Requeriments no funcionals

- L'aplicació ha d'estar desenvolupada en Java.
 - El motor de la base de dades ha de ser MySQL.
-

- Sistema escalable, robust i ràpid.
- El sistema operatiu que ha de suportar ha de ser Windows.
- S'ha de treballar de manera àgil i aplicant el Patró Model Vista Controlador 4 capes.
- La interfície ha de ser senzilla, còmode i intuïtiva.
- La interacció amb l'usuari ha de ser fluida.

4.3. Diagrama de casos d'ús

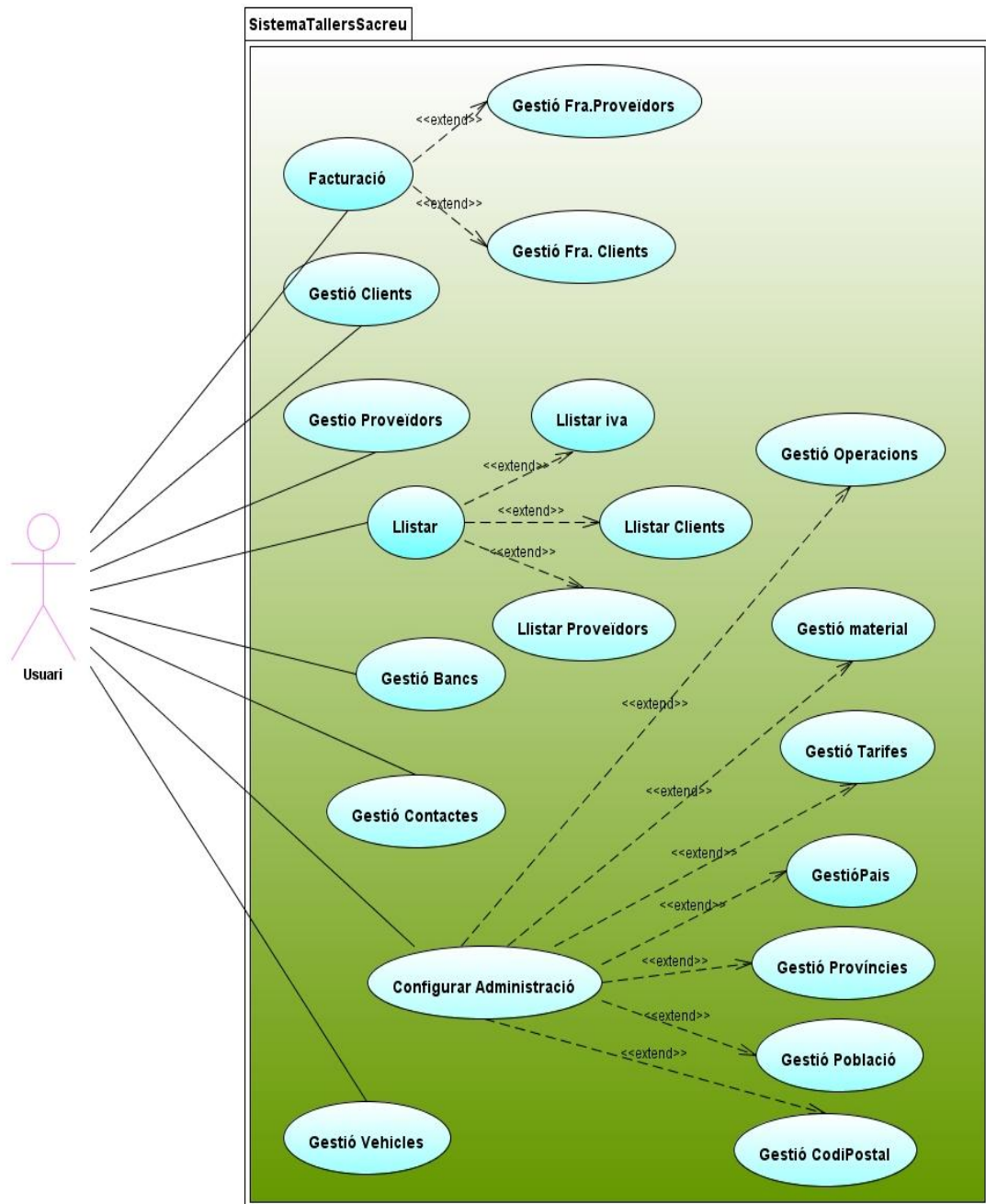


Figura 2. Diagrama de casos d'ús

- **Gestió Client:** Consta de tres casos d'ús: Alta Client, Baixa Client i Modificació Client.
 - **Gestió Proveïdor:** Consta de tres casos d'ús: Alta Proveïdor, Baixa Proveïdor i Modificació Proveïdor.
 - **Facturació:** Engloba dos casos:
 - **Gestió Facturació Proveïdors:** Consta de tres casos d'ús: Alta Factura de Proveïdor, Baixa Factura de Proveïdor, Modificació Factura Proveïdor.
 - **Gestió Factura Clients:** Consta de dos casos d'ús: Alta Factura Clients i Modificació Factura Proveïdors.
 - **Gestió Contactes:** Consta de tres casos d'ús: Alta Contacte, Baixa Contacte, Modificació Contacte.
 - **Gestió Bancs:** Consta de tres casos d'ús: Alta Compte Bancari, Baixa Compte Bancari, Modificació Compte Bancari.
 - **Gestió Vehicles:** Hi ha 3 casos d'ús, Alta Vehicle, Baixa Vehicle, Modificació Vehicle.
 - **Llistar:** En aquest cas d'ús es creen diferents llistes:
 - **Llistar iva:** Creació de llistes d'ives mensuals, tant de proveïdors com de clients.
 - **Llistar Clients:** Es crea la llista del clients de l'empresa Talleres Sacreu s.c.c.l.
 - **Llistar Proveïdors:** Crea la llista de proveïdors relacionat amb Tallers Sacreu s.c.c.l.
 - **Gestió Administració:** s'hi engloba tot el que fa referència a la configuració del dades del sistema, es dir, les dades de configuració de l'usuari del sistema, sense aquestes dades introduïdes el sistema no pot executar la majoria de casos d'ús. S'ha fet aquests casos d'us per evitar errors d'introducció de dades per part d l'usuari. Aquest cas d'ús és que més n'engloba, fins a un total de set.
 - **Gestió Tarifes:** Configura els diferents preus preus/hora dels que disposa el taller. Com les altres gestions consta de d'Alta Tarifa, Modificar Tarifa i Baixa Tarifa.
 - **Gestió País:** Consta d'Alta País i Baixa País.
 - **Gestió Províncies:** Consta de 2 casos d'ús Alta País i Baixa País.
 - **Gestió Població:** Com les anteriors, Alta població i Modificar Població.
-

- **Gestió Codis Postals:** consta de Alta Codi Postal i Baixa Codi Postal.
- **Gestió Operacions:** Alta i Modificació Operacions.
- **Gestió Material:** Alta Material i baixa Material.

Aquest últims 6 casos són únicament per evitar duplicacions errònies a la base de dades. Per exemple, si ja hi ha les poblacions a dins de la base de dades, s'evita que l'usuari un cop insereixi "Mataró" i un altre cop "Mataro" (una vegada amb accent i una altra sense).

4.4. Especificació de casos d'ús

4.4.1. Gestió Clients

Cas d'ús "Alta Client"

Actor/s: Usuari.

El cas d'ús pot finalitzar sempre que l'usuari ho desitgi.

Pre-Condicion

1. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).

Post-Condicion

1. El client que enregistra.
2. Si hi ha dades del contacte, contacte enregistra.
3. Si hi ha dades del banc, banc enregistra.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Client".
 2. El sistema demana el "Nif" del nou client.
 3. L'usuari introdueix el "Nif".
-

4. L'usuari confirma la introducció del "Nif".
5. El sistema comprova el "Nif".
6. El sistema mostra el formulari per introduir les dades fiscal, els contactes i el número de compte del client. Les dades demanades són: identificador de client, data d'alta, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), nif, direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses) , tarifa aplicada,nom del contacte,telèfons, telèfons mòbils,faxs, correus electrònics, nom banc, entitat, oficina, dígit de control, numero de compte .
7. L'usuari introdueix les dades del client (identificador de client, data d'alta, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses) , tarifa aplicada), en cas de que tingui algun tipus d'informació del contacte(telèfons, telèfons mòbils, faxs o correus electrònics) també la introdueix, i finalment si té en numero de compte del contacte també el pot introduir (nom del banc, entitat, oficina, dígit de control i numero de compte).
8. L'usuari confirma les dades introduïdes prement el boto "acceptar"
9. El sistema valida les dades.
10. El sistema necessita confirmació de que les dades son correctes.
11. L'usuari prem "Sí".
12. El sistema demana si es volen introduir vehicles del nou client.
13. L'usuari prem "Sí".
14. El sistema inicia el cas d'ús "Alta vehicle" des del punt 2.
15. Finalitza el cas d'ús "Alta Client".

Flux Alternatiu

- 4.1. El sistema detecta que el "Nif" es incorrecte.
 - 4.1.1. Retorna al punt 2.
 - 4.2. El sistema detecta que el "Nif" esta repetit.
 - 4.2.1 Retorna al punt 2.
 - 4.3. El sistema detecta que el "Nif" pertany a un proveïdor.
-

- 4.3.1. El sistema demana si es volen copiar les dades fiscals del proveïdor al nou client.
- 4.3.2. Es va directament al punt 6, on les dades ja estan en el formulari per si se'n desitja modificar alguna.
- 4.4. El sistema detecta que el "Nif" és d'un client donat de baixa.
 - 4.4.1. El sistema demana si es vol canviar l'estat del client, d'estar donat de baixa a estar donat d'alta.
 - 4.4.2. S'inicia el cas d'ús "Modificar Client" en el punt 4, on les dades ja estan en el formulari per si se'n desitja modificar alguna.
 - 4.4.3. Finalitza el cas d'ús "Alta Client".
- 7.1. El sistema detecta que hi ha alguna dada incorrecte.
 - 7.1.1. Retorna al pas 7.
- 11.1. L'usuari prem "No".
 - 11.1.1. El sistema retorna fins al punt 8.
- 13.1. L'Usuari prem "No".
 - 13.1.1 Finalitza el cas d'ús.

Cas d'ús "Modificar Client"

Actor: Usuari.

El cas d'ús pot finalitzar l'usuari desitgi.

Sense sortir del cas d'ús es pot saltar cap al cas d'ús "Modificar Contactes", "Modificar Bancs" i "Modificar Vehicles" però només del client seleccionat.

Pre-Condicions

1. Tenir almenys un client donat a la base de dades.
2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).

Post-Condicions

1. Emmagatzemar les noves dades del client.
-

Flux Normal

1. L'usuari inicia el cas d'ús "Modificar Client".
2. El sistema mostra el llistat de tots els clients emmagatzemats a base de dades tant actius com donats de baixa.
3. L'usuari selecciona el client desitjat de llista i accepta.
4. El sistema mostra el formulari amb les següents dades del client que es vol modificar. Dades que mostra: identificador de client, data d'alta, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses) , tarifa aplicada i si esta actiu o donat de baixa.
5. L'usuari modifica les dades que necessiti canviar, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses) , tarifa aplicada i si esta actiu o donat de baixa.
6. L'usuari confirma les dades introduïdes prement el boto "acceptar"
7. El sistema valida les dades.
8. El sistema necessita confirmació de que les dades son correctes.
9. L'usuari prem "Sí".
10. El sistema realitza l'acció.
11. El sistema pregunta si es desitja sortir del cas d'ús o si es volen modificar dades dels contactes o dels bancs.
12. L'usuari prem "No".
13. El sistema surt del cas d'ús.

Flux alternatiu

- 7.1. El sistema detecta errors en les noves dades introduïdes.
 - 7.1.1. Retorn al pas 5.
 - 9.1. L'usuari prem "No".
 - 9.1.1. Retorn al pas 5.
 - 12.1. L'usuari prem "Sí".
-

- 12.1.1. El sistema pregunta quines dades es volen modificar, mostrant 4 opcions, “Dades Contacte”, “Dades Banc”, “Dades Vehicle”, “Cancel·lar”, si el client no té alguna de les dades el botó està desactivat .
- 12.1.2. L’usuari fa la selecció d’alguna de les opcions a les quals tingui accés.
- 12.1.3. El sistema inicia el cas d’ús referent a l’opció escollida per l’usuari.

Cas d’ús “Baixa Client”

Actor: Usuari.

El cas d’ús pot finalitzar quan l’usuari ho desitgi.

Pre-Condicions

1. Tenir clients actius a la base de dades.
2. El clients que es desitgin donar de baixa no poden tenir factures en els últims 5 anys, per raons fiscals.

Post-Condicions

1. Client donat de baixa a la base de dades.
2. Si es dóna el cas, i té contactes o bancs que no comparteixi amb cap proveïdor també es donen de baixa.
3. Si té algun vehicle, aquest també es dóna de baixa.
4. Si té alguna factura posterior a 5 anys també s’elimina, permanentment de la base de dades.

Flux normal

1. L’usuari inicia el cas d’ús “Donar de Baixa Client”.
 2. El sistema mostra un llistat del clients actius.
 3. L’usuari selecciona el client que desitja eliminar i confirma la selecció.
 4. El sistema comprova que no té cap factura en els últims cinc anys.
-

5. El sistema mostra les totes les dades del client, tots el contactes relacionats, tots els vehicles relacionats i les dades bancàries .
6. L'usuari prem el botó Acceptar.
7. El sistema pregunta si esta segur de voler donar de baixa el client seleccionat.
8. L'usuari prem "Sí".
9. El sistema dóna de baixa el client i si es dona el cas també dóna de baixa els seus contactes , les seves dades bancàries i els seus vehicles, sempre i quan no estiguin relacionats amb un proveïdor.
10. Finalitza el cas d'ús.

Flux alternatiu

- 4.1. El sistema detecta que existeixen factures en els últims cinc anys del client.
 - 4.1.1. El sistema informa al usuari de la incidència.
 - 4.1.2. Surt del cas d'ús.
- 8.1. L'usuari prem "No".
 - 8.1.1. Surt del cas d'ús.

4.4.2. Gestió Proveïdors

Cas d'ús "Alta Proveïdor"

Actor: Usuari

El cas d'ús pot finalitzar sempre que l'usuari ho desitgi.

Pre-Condicions

1. Paràmetres d'administració del sistema introduïts al sistema (països, províncies i poblacions).

Post-Condicions

1. El proveïdor queda enregistrat.
-

2. Si hi ha dades del contacte, contacte enregistrat.
3. Si hi ha dades del banc, banc enregistrat.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Proveïdor".
 2. El sistema demana el "Nif" del nou client.
 3. L'usuari introdueix el "Nif".
 4. L'usuari confirma la introducció del "Nif".
 5. El sistema comprova el "Nif".
 6. El sistema mostra el formulari per introduir les dades del Proveïdor. Les dades fiscals del proveïdor són: identificador de proveïdor, data d'alta, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses), nom del contacte, telèfons, telèfons mòbils, faxes, correus electrònics, nom banc, entitat, oficina, dígit de control, numero de compte .
 7. L'usuari introdueix les dades del proveïdor,(identificador de proveïdor, data d'alta, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses)) i si existeixen també introdueix les dades referents a contacte (nom del contacte, telèfons, telèfons mòbils, faxes, correus electrònics) i al banc (nom banc, entitat, oficina, dígit de control, numero de compte).
 8. L'usuari confirma les dades introduïdes prement el boto "acceptar"
 9. El sistema valida les dades.
 10. El sistema requereix confirmació de que les dades introduïdes son correctes.
 11. L'usuari prem "Sí".
 12. Si hi ha alguna dada del contacte del proveïdor ,el sistema demana si es volen introduir més contactes del nou proveïdor.
 13. L'usuari prem "No".
 14. Finalitza el cas d'ús "Alta Proveïdor"
-

Flux Alternatiu

- 4.1. El sistema detecta que el “Nif” es incorrecte.
 - 4.1.2. Retorna al punt 2.
- 4.2. El sistema detecta que el “Nif” esta repetit.
 - 4.2.1 Retorna al punt 2.
- 4.3. El sistema detecta que el “Nif” pertany a un client.
 - 4.3.1. El sistema demana si es volen copiar les dades personals del client al nou proveïdor.
 - 4.3.2. Es va directament al punt 6, on les dades ja estan en el formulari.
- 11.1. L’usuari prem “No”.
 - 11.1.1. El sistema retorna al pas 8.
- 13.1. L’usuari prem “Sí”
 - 13.1.1. El sistema inicia el cas d’ús “Alta Contacte” des del punt 4.

Cas d’ús “Modificar Proveïdor”

Actor: Usuari

L’usuari pot sortir del cas d’ús quan ho desitgi.

Pre-Condicions

1. Tenir almenys un proveïdor donat a la base de dades.
2. Paràmetres d’administració del sistema introduïts al sistema (països, províncies i poblacions).

Post-Condicions

1. Emmagatzemar les noves dades del proveïdor.

Flux Normal

1. L’usuari inicia el cas d’ús “Modificar Proveïdor”.
-

2. El sistema mostra el llistat de tots els proveïdors emmagatzemats a base de dades tant actius com donats de baixa.
3. L'usuari selecciona el proveïdor desitjat de llista i accepta.
4. El sistema mostra el formulari amb les dades del proveïdor que es vol modificar. Dades que mostra: identificador de proveïdor, data d'alta, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses) i si esta actiu o donat de baixa.
5. L'usuari modifica les dades que necessiti canviar, nom i cognoms (en cas de persones), nom comercial i nom fiscal (en cas d'empreses), direcció postal , direcció fiscal, país, província, població, codi postal, web (en cas d'empreses) , i si esta actiu o donat de baixa.
6. L'usuari confirma les dades introduïdes prement el boto "acceptar"
7. El sistema valida les dades.
8. El sistema necessita confirmació de que les dades son correctes.
9. L'usuari prem "Sí".
10. El sistema realitza l'acció.
11. El sistema pregunta si es desitja sortir del cas d'ús o si es volen modificar dades dels contactes o dels bancs.
12. L'usuari prem "No".
13. El sistema surt del cas d'ús.

Flux Alternatiu

- 7.2. El sistema detecta errors en les noves dades introduïdes.
 - 7.2.1. Retorn al pas 5.
 - 9.2. L'usuari prem "No".
 - 9.2.1. Retorn al pas 5.
 - 12.2. L'usuari prem "Sí".
 - 12.2.1. El sistema pregunta quines dades es volen modificar, mostrant 3 opcions, "Dades Contacte" , "Dades Banc", "Cancel·lar", si el client no té alguna de les dades el botó està desactivat .
 - 12.2.2. L'usuari fa la selecció d'alguna de les opcions a les quals tingui accés.
-

12.2.3. El sistema inicia el cas d'ús referent a l'opció escollida per l'usuari.

Cas d'ús "Baixa Proveïdor"

Actor: Usuari.

El cas d'ús pot finalitzar quan l'usuari ho desitgi.

Pre-Condicions

1. Tenir proveïdors actius a la base de dades.
2. El proveïdors que es desitgin donar de baixa no poden tenir factures relacionades en els últims 5 anys, per raons fiscals.

Post-Condicions

1. Proveïdor donat de baixa a la base de dades.
2. Si es dóna el cas, i té contactes que no comparteix amb cap client, també es donen de baixa.
3. Si es dóna el cas, i té bancs que no comparteixi amb cap client també es donen de baixa.
4. Si té alguna factura posterior a 5 anys també s'elimina, permanentment de la base de dades.

Flux normal

1. L'usuari inicia el cas d'ús "Donar de Baixa Proveïdor".
 2. El sistema mostra un llistat dels proveïdors actius.
 3. L'usuari selecciona el proveïdor que desitja eliminar i confirma la selecció.
 4. El sistema comprova que no té cap factura en els últims cinc anys.
 5. El sistema mostra les totes les dades del proveïdor, totes les dades dels contactes i dels bancs relacionats amb dit client.
 6. L'usuari prem el botó Acceptar.
-

7. El sistema pregunta si esta segur de voler donar de baixa el proveïdor seleccionat.
8. L'usuari prem "Sí".
9. El sistema dona de baixa el proveïdor i si es dona el cas també dona de baixa, contactes i bancs, sempre i quan no estiguin relacionades amb un client.
10. Finalitza el cas d'ús.

Flux alternatiu

- 4.2. El sistema detecta que existeixen factures en els últims cinc anys del proveïdor
 - 4.2.1. El sistema informa al usuari de la incidència.
 - 4.2.2. Surt del cas d'ús.
- 8.2. L'usuari prem "No".
 - 8.2.1. Surt del cas d'ús.

4.4.3. Gestió Contactes

Cas d'ús "Alta Contacte"

Actor: Usuari.

El cas d'ús pot finalitzar sempre que l'usuari ho desitgi.

L'usuari pot passar en qualsevol moment al cas d'ús "Modificar Contacte".

Pre-Condicions

1. Tenir algun client o proveïdor actiu (que no estigui donat de baixa) enregistrat.
2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).

Post-Condicions

1. Contacte enregistrat, amb telèfons, mòbils, faxes i/o correus electrònics.
-

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Contacte".
2. El sistema mostra tots el clients i proveïdors actius emmagatzemats.
3. L'usuari selecciona el client o proveïdor.
4. El sistema mostra una llista amb els contactes existents per al client o proveïdor seleccionat i el formulari per introduir les dades del nou Contacte, el formulari demana nom del contacte, telèfons, mòbils, faxes, e-mails .
5. L'usuari introdueix les dades del contacte, el nom, i almenys una dada de pertanyent al client, telèfon o mòbil o fax o e-mail.
6. L'usuari confirma les dades introduïdes prement el boto "acceptar"
7. El sistema valida les dades.
8. El sistema confirma l'alta i demana si es volen introduir més contactes.
9. L'usuari prem el boto "No".
10. Finalitza el cas d'ús "Alta Contacte".

Flux Alternatiu

- 1.1. El sistema detecta que no hi cap client ni cap proveïdor actiu guardat.
 - 4.1.2. Surt del cas d'ús.
- 6.1. El sistema detecta dades incorrectes
 - 6.1.1 Retorna al punt 4 .
- 9.1. L'usuari prem "Sí".
 - 9.1.1. El sistema retorna al pas 2.

Cas d'ús "Modificar Contacte"

Actor: Usuari

L'usuari pot sortir del cas d'ús quan ho desitgi.

Pre-Condicions

1. Tenir almenys un proveïdor o contacte amb contactes relacionats, a la base de dades.
-

2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).

Post-Condicions

1. Emmagatzemar les noves dades del client (telèfons, telèfons mòbils, faxes, correus electrònics).

Flux Normal

1. L'usuari inicia el cas d'ús "Modificar Contacte".
2. El sistema mostra el llistat de tots els proveïdors i clients emmagatzemats a base de dades tant actius com donats de baixa i que disposen de dades de contactes.
3. L'usuari selecciona el proveïdor o client desitjat de llista i accepta.
4. El sistema mostra el formulari amb les dades del contacte que es vol modificar. Dades que mostra: Nom del contacte, telèfons, telèfons mòbils, faxes i e-mails i si esta actiu o donat de baixa.
5. L'usuari modifica les dades que necessiti canviar, telèfons, telèfons mòbil, faxes, correus electrònics i si esta actiu o donat de baixa.
6. L'usuari confirma les dades introduïdes prement el boto "acceptar".
7. El sistema valida les dades.
8. El sistema necessita confirmació de que les dades són correctes.
9. L'usuari prem "Sí".
10. El sistema realitza l'acció.
11. El sistema surt del cas d'ús.

Flux Alternatiu

7.3. El sistema detecta errors en les noves dades introduïdes.

7.3.1. Retorn al pas 5.

9.3. L'usuari prem "No".

9.3.1. Retorn al pas 5.

Cas d'ús "Baixa Contacte"

Actor:Usuari.

Pre-Condicions

1. Tenir clients o proveïdors amb dades de contactes, a la base de dades.
2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).

Post-Condicions

1. Eliminació permanent d'un contacte de la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Eliminar Contacte".
2. El sistema mostra el llistat de tots els proveïdors i clients actius emmagatzemats a base de dades i que disposen de dades de contactes.
3. L'usuari selecciona el proveïdor o client desitjat de llista i accepta.
4. El sistema mostra un llistat amb contactes del client o proveïdor seleccionat.
5. L'usuari selecciona i accepta el contacte que vol eliminar.
6. El sistema comprova que el contacte seleccionat només té relació amb el client o proveïdor seleccionat anteriorment.
7. El sistema necessita confirmació de que es vol eliminar de forma permanent les dades del contacte, nom, telèfons, telèfons mòbils, faxes i correus electrònics.
8. L'usuari prem "Sí".
9. El sistema realitza l'acció.
10. El sistema surt del cas d'ús.

Flux Alternatiu

- 7.1. El sistema comprova que te relació amb altres proveïdors o clients, i pregunta si es vol eliminar de totes maneres.
-

7.1.1. L'usuari prem "Sí" .

7.1.1.1. Avança al pas 9.

7.1.2. L'usuari prem "No".

7.1.2.1. Surt del cas d'ús.

4.4.4. Gestió Vehicles

Cas d'ús "Alta Vehicle"

Actor: Usuari.

El cas d'ús pot finalitzar sempre que l'usuari ho desitgi.

Pre-Condicions

1. Tenir algun client actiu (que no estigui donat de baixa) enregistrat.
2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).

Post-Condicions

1. El vehicle queda enregistrat.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Vehicle".
 2. El sistema mostra tots el clients actius emmagatzemats.
 3. L'usuari selecciona el client.
 4. El sistema mostra les dades del client seleccionat, dels seus contactes i dels seus vehicles existents i el formulari per introduir les dades del nou Vehicle.
 5. L'usuari introdueix les dades del contacte.
 6. L'usuari confirma les dades introduïdes prement el boto "acceptar"
 7. El sistema valida les dades.
-

8. El sistema confirma l'alta i demana si es volen introduir més vehicles del mateix client.
9. L'usuari prem el boto "No".
10. Finalitza el cas d'ús "Alta Vehicle".

Flux Alternatiu

- 4.1. El sistema detecta que no hi cap client actiu guardat.
 - 4.1.2. Surt del cas d'ús.
- 6.1. El sistema detecta dades incorrectes
 - 6.1.1. Retorna al punt 4 .
- 9.1. L'usuari prem "Sí".
 - 9.1.1. El sistema retorna al pas 4, actualitzat.

Cas d'ús "Modificar Vehicle"

Actor: Usuari.

Sempre que es desitgi es pot abandonar el cas d'ús.

Pre-Condicions

1. Tenir clients actius amb vehicles emmagatzemats a la base de dades.
2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).

Post-Condicions

1. Tenir emmagatzemades les noves dades del vehicle seleccionat.

Flux Normal

1. L'usuari inicia el cas d'ús "Modificar Vehicle".
-

2. El sistema mostra tots el clients actius amb dades de vehicles, que estan emmagatzemats a la base de dades.
3. L'usuari selecciona un client i accepta.
4. El sistema mostra la informació del client, i la llista dels vehicles que té.
5. L'usuari selecciona un vehicle.
6. El sistema mostra les dades del vehicle seleccionat en un formulari. Les dades són: matrícula, tipus, marca, model, data d'alta i quilòmetres.
7. L'usuari canvia les dades antigues per les noves, tipus, marca, model i quilòmetres i la data de baixa en cas de tenir-ne i accepta.
8. El sistema valida les dades.
9. El sistema pregunta si les dades són correctes.
10. L'usuari ho confirma.
11. Finalitza el cas d'ús.

Flux Alternatiu

- 7.1. El sistema detecta algun error en les noves dades.
 - 7.1.1. Retorna al pas 6.
- 10.1. L'usuari prem "No".
 - 10.1.1. Retorna al pas 6.

Cas d'ús "Baixa Vehicle"

Actor:Usuari.

Sempre que es desitgi es pot abandonar el cas d'ús.

Pre-Condicion

1. Tenir clients actius amb vehicles actius emmagatzemats a la base de dades.
 2. Que els vehicles actius no tinguin cap factura en els últims 5 anys
 3. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies i poblacions).
-

Post-Condicions

1. Tenir el vehicle seleccionat donat de baixa a la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Donar de Baixa Vehicle".
2. El sistema mostra tots el clients actius amb dades de vehicles actius, que estan emmagatzemats a la base de dades.
3. L'usuari selecciona un client i accepta.
4. El sistema mostra la informació del client, i la llista dels vehicles actius que té.
5. L'usuari selecciona un vehicle de la llista.
6. El sistema comprova que no té cap factura relacionada.
7. El sistema mostra les dades del vehicle en un formulari. Les dades són: matrícula, tipus, marca, model, data d'alta i quilòmetres.
8. L'usuari prem "Acceptar".
9. El sistema valida les dades.
10. El sistema pregunta si esta segur de voler donar de baixa el vehicle.
11. L'usuari ho confirma.
12. Finalitza el cas d'ús.

Flux Alternatiu

- 6.1. El sistema comprova que té factures amb una antiguitat menor a 5 anys.
 - 6.1.1. El sistema surt del cas d'ús.
- 10.1. L'usuari no confirma la baixa
 - 10.1.1. El sistema surt del cas d'ús.

4.4.5. Gestió Factures Proveïdors

Cas d'ús "Alta Factures Proveïdors"

Actor:Usuari.

L'usuari decideix quan aturar el cas d'ús.

Pre-Condicions

1. Tenir proveïdors actius donats d'alta a la base de dades

Post-Condicions

2. Tenir emmagatzemada una factura de proveïdor a la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Factures Proveïdors".
 2. El sistema mostra el formulari per introduir les factures noves. El formulari consta dels següents camps: mes, any, número factura taller, identificador del proveïdor, nom fiscal del proveïdor, data d'emissió de la factura, numero factura proveïdor, base imposable, iva, irpf, total factura, tant per cent iva i tant per cent irpf.
 3. L'usuari selecciona el mes i introdueix l'any.
 4. El sistema comprova que el mes i l'any son iguals o anteriors a la data actual i també comprova si hi ha factures del mes de any seleccionat i mostra les dades de les factures existents en una taula.
 5. L'usuari introdueix les dades necessàries en els camps corresponents, selecciona el proveïdor, la data d'emissió i comprova que els tants per cent d'iva i irpf corresponen amb la seva factura i finalment introdueix la base imposable.
 6. El sistema valida de les dades i fa els càlculs de l'iva, irpf i total factura corresponents i ho mostra en el formulari d'introducció de dades.
 7. L'usuari prem el boto "Afegir a la llista".
-

8. El sistema insereix les dades de la factura a la base de dades i afegeix les dades a la taula del formulari.
9. El cas d'ús finalitza quan l'usuari decideix acabar d'inserir factures.

Flux Alternatiu

- 4.1. El sistema detecta que el mes i l'any introduïts són posteriors a la data actual.
 - 4.1.1. Retorn al pas 2.
- 6.1. El sistema detecta que el valor base imposable és incorrecte.
 - 6.1.1. Retorn al pas 5.

Cas d'ús "Modificar Factura Proveïdor"

Actor:Usuari.

L'usuari decideix quan aturar el cas d'ús.

Pre-Condicions

1. Tenir proveïdors actius donats d'alta a la base de dades.
2. Tenir factures de proveïdors del mes i any seleccionats.

Post-Condicions

1. Tenir les noves dades d'una factura de proveïdor a la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Modificar Factures Proveïdors".
 2. El sistema mostra el formulari per introduir les factures noves. El formulari consta dels següents camps: mes, any, número factura taller, identificador del proveïdor, nom fiscal del proveïdor, data d'emissió de la factura, numero factura proveïdor, base imposable, iva, irpf, total factura, tant per cent iva i tant per cent irpf.
-

3. L'usuari selecciona el mes i introdueix l'any.
4. El sistema comprova que el mes i l'any son iguals o anteriors a la data actual i també comprova si hi ha factures del mes de any seleccionat i mostra les dades de les factures existents en una taula.
5. L'usuari selecciona la factura que es vol modificar
6. El sistema omple els camps del formulari amb les dades de la factura seleccionada.
7. L'usuari modifica les dades antigues per les dades noves.
8. El sistema valida de les dades i fa els càlculs de l'iva, irpf i total factura corresponents i ho mostra en el formulari d'introducció de dades.
9. L'usuari prem el boto "Modificar".
10. El sistema modifica les dades de la factura a la base de dades i també modifica les dades a la taula del formulari.
11. El cas d'ús finalitza quan l'usuari decideix acabar d'inserir factures.

Flux Alternatiu

- 4.1. El sistema comprova que no té cap factura el mes i l'any introduïts.
 - 4.1.1. Retorn al pas 2.
- 6.1. El sistema detecta que el valor base imposable és incorrecte.
 - 6.1.1. Retorn al pas 5.

4.4.6. Gestió Factures Clients

Cas d'ús "Alta Factures Clients"

Actor:Usuari.

L'usuari decideix quan aturar el cas d'ús.

Pre-Condicions

1. Tenir clients actius donats d'alta a la base de dades, amb altes d'ordre sense relacionar amb cap factura.
-

Post-Condicions

1. Tenir emmagatzemada una factura de clients a la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Factures Proveïdors".
2. El sistema mostra els clients actius que tenen ordres de taller pendents de facturar.
3. L'usuari selecciona un client.
4. El sistema mostra les dades del client, les ordres de taller pendents de facturar i el formulari per introduir les factures noves. El formulari consta dels següents camps: número factura, data d'emissió, data de venciment.
5. L'usuari introdueix les dades i accepta.
6. El sistema valida les dades, mostra la informació la factura que es crearà i necessita confirmar les dades.
7. L'usuari confirma les dades.
8. El sistema pregunta si es volen fer més factures.
9. L'usuari prem "No"
10. Finalitza el cas d'ús.

Flux Alternatiu

- 6.1. El sistema detecta alguna dada incorrecta.
 - 6.1.1. Retorn al pas 5.
 - 6.2. El sistema detecta que la data de venciment és posterior a la data d'emissió de la factura .
 - 6.2.1. Retorn al pas 5.
 - 6.3. El sistema detecta que el client té un banc actiu, i detecta que la dia de venciment no és 5 ,ni 20, 25 del mes següent a la emissió.
 - 9.1. L'usuari prem "Sí".
 - 9.1.1. Retorn al pas 2.
-

Cas d'ús “Modificar Factura Client”

Actor:Usuari.

L'usuari decideix quan aturar el cas d'ús.

Pre-Condicions

1. Tenir clients actius donats d'alta a la base de dades, amb vehicles.
2. Tenir ordres de taller del vehicle del client.
3. Tenir una factura del client a la base de dades.
4. No haver tancat el llistat d'ives de clients del mes d'emissió de la factura.

Post-Condicions

1. Tenir les noves dades d'una factura de client a la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús “Modificar Factures Client”.
2. El sistema mostra les factures existents.
3. L'usuari selecciona la factura que vol modificar.
4. El sistema mostra el formulari amb les dades de la factura seleccionada.
5. L'usuari modifica les dades i les ordres de taller seleccionades.
6. El sistema comprova les noves dades introduïdes.
7. El sistema valida les dades, mostra la informació la factura que es modificarà i necessita confirmar les dades.
8. L'usuari confirma.
9. El sistema realitza l'acció i finalitza el cas d'ús.

Flux Alternatiu

- 6.1. El sistema detecta alguna dada incorrecta.
 - 6.1.1. Retorn al pas 5.
-

6.2. El sistema detecta que la data de venciment és posterior a la data d'emissió de la factura .

6.2.1. Retorn al pas 5.

6.3. El sistema detecta que el client té un banc actiu, i detecta que la dia de venciment no és 5 ,ni 20, 25 del mes següent a la emissió.

4.4.7. Gestió de bancs

Cas d'ús "Alta Banc"

Actor: Usuari

L'usuari pot abandonar el cas d'ús quan ho desitgi.

Pre-condicions

1. Tenir proveïdors o clients actius emmagatzemats a la base de dades.

Post-Condicions

1. Tenir registrat un banc relacionat amb un client i/o un proveïdor actius.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Banc"
 2. El sistema mostra el llistat de clients i proveïdors actius.
 3. L'usuari selecciona un client o proveïdor i accepta.
 4. El sistema mostra les dades del client o proveïdor, i si té algun banc relacionat també n'ensenya les dades del banc en una taula. També mostra el formulari d'entrada on es demana el nom, l'entitat,l'oficina, el dígit de control i el numero de compte.
 5. L'usuari introdueix el nom del banc, l'entitat, l'oficina, el dígit de control i el numero de compte i accepta.
 6. El sistema valida les dades i demana si les dades del banc que s'acaben d'introduir són les noves dades per defecte del bancs del client
-

7. L'usuari confirma.
8. El sistema realitza l'acció i surt del cas d'ús.

Flux Alternatiu

- 6.1. El sistema detecta que les dades són incorrectes
 - 6.1.1. Retorn al pas 4.
- 6.2. El sistema detecta que el banc es repeteix en un altre client.
 - 6.2.1. Retorn al pas 4
- 7.1. L'usuari prem "No".
 - 7.1.1. El sistema continua l'acció però sense posar per defecte les dades del banc.

Cas d'ús "Modificar Banc"

Actor: Usuari

L'usuari pot sortir del cas d'ús quan ho desitgi.

Pre-Condicions

1. Tenir a la base de dades clients o proveïdors actius que tinguin bancs relacionats.

Post-Condicions

1. Enregistrar les noves dades del banc a la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Modificar Banc"
 2. El sistema mostra el llistat de clients i proveïdors actius amb bancs relacionats.
 3. L'usuari selecciona un client o proveïdor de la llista.
 4. El sistema mostra el bancs del clients en una taula.
 5. L'usuari selecciona un banc.
-

6. El sistema introdueix les dades del banc seleccionat al formulari de la finestra, nom banc, entitat, oficina, dígit de control, numero de compte i si és l'actual.
7. L'usuari modifica les dades necessàries i accepta.
8. El sistema valida les dades i demana si vol que el banc sigui l'actual.
9. L'usuari prem "Sí".
10. El sistema fa l'acció i finalitza el cas d'ús.

Flux Alternatiu

- 8.1. El sistema detecta que les dades són incorrectes
 - 8.1.1. Retorn al pas 4.
- 8.2. El sistema detecta que el banc es repeteix en un altre client.
 - 8.2.1. Retorn al pas 4.
- 9.1. L'usuari prem "No".
 - 9.1.1. El sistema continua l'acció però sense posar per defecte les dades del banc.

Cas d'ús "Eliminar Banc"

Actor: Usuari.

L'usuari pot sortir del cas d'ús quan ho desitgi.

Pre-Condicions

1. Tenir contactes clients o proveïdors actius amb alguna dada de banc.

Post-Condicions

1. Eliminació permanent del banc seleccionat a la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Eliminar Banc"
 2. El sistema mostra el llistat de clients i proveïdors actius amb bancs relacionats.
-

3. L'usuari selecciona un client o proveïdor de la llista.
4. El sistema mostra el bancs del clients en una taula.
5. L'usuari selecciona un banc.
6. El sistema demana si es vol eliminar permanentment de la base de dades
7. L'usuari prem "Sí".
8. El sistema fa l'acció i finalitza el cas d'ús.

Flux Alternatiu

7.1. L'usuari prem "No".

7.1.1. Retorn al pas 4.

4.4.8. Gestió Ordres de Taller

Cas d'ús "Alta Ordre Taller"

Autor: Usuari

L'usuari pot sortir del cas d'ús quan ho desitgi.

Pre-Condicions

1. Tenir clients actius que posseeixin vehicles que no estiguin donats de baixa.
2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies, poblacions operacions).

Post-Condicions

1. Tenir a la base de dades una ordre de taller del vehicle seleccionat.

Flux Normal

1. L'usuari inicia el cas d'ús alta ordre taller.
 2. El sistema mostra un llistat del clients actius que tenen vehicles.
-

3. L'usuari selecciona un client.
4. El sistema mostra les dades del client, les dades de totes els contactes i les dades de tots els vehicles.
5. L'usuari selecciona el vehicle desitjat.
6. El sistema mostra les dades del vehicle i el formulari per inserir les següents dades: nom operació, nom material, quantitat i preu unitari.
7. L'usuari introdueix les dades data inici, nom operació, nom material, quantitat, preu unitari, quilòmetres del vehicle el dia d'entrada al taller.
8. El sistema valida les dades, calcula el camp preu total i afegeix les operacions i les ordres a les llistes corresponents.
9. L'usuari prem "Finalitzar"
10. El sistema demana el total d'hores de durada de l'ordre de taller.
11. L'usuari entra les hores totals.
12. Els sistema valida les dades.
13. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 8.1. El sistema detecta dades incorrectes o repetides
 - 8.1.1. Retorn al pas 7.
- 9.1. L'usuari no prem "Finalitzar".
 - 9.1.1. Retorn al pas 7.
- 12.1. El sistema detecta que el valor de les hores totals es incorrecte.
 - 12.1.1. Retorn al pas 11.

Cas d'ús "Modificar Ordre Taller"

Autor: Usuari

L'usuari pot sortir del cas d'ús quan ho desitgi.

Pre-Condicions

1. Tenir clients actius que posseeixin vehicles que no estiguin donats de baixa.
-

2. Paràmetres d'administració del sistema introduïts al sistema (Tipus de tarifes, països, províncies, poblacions, operacions, material).

Post-Condicions

1. Tenir a la base de dades una ordre de taller del vehicle seleccionat.

Flux Normal

1. L'usuari inicia el cas d'ús "modificar ordre taller".
2. El sistema mostra un llistat del clients actius que tenen vehicles.
3. L'usuari selecciona un client.
4. El sistema mostra les dades del client, les dades de totes els contactes i les dades de tots els vehicles.
5. L'usuari selecciona el vehicle desitjat.
6. El sistema mostra les dades del vehicle i les ordres de taller pendents de facturar.
7. L'usuari escull la ordre de taller desitjada.
8. El sistema mostra la informació de la ordre selecciona al formulari.
9. L'usuari modifica les operacions que si li ha realitzat en vehicle, i/o els materials i accepta.
10. El sistema valida les dades i demana confirmació.
11. L'usuari confirma que les dades són correctes.
12. El sistema finalitza el cas d'ús.

Flux alternatiu

- 10.1. El sistema detecta dades incorrectes.
 - 10.1.1. Retorn al pas 9.
- 10.2. L'usuari no confirma.
 - 10.2.1. Retorn al pas 9.

4.4.9. Gestió Tarifes

Cas d'ús "Alta Tarifa Preu"

Actor:Usuari.

Pre-Condicions

Post-Condicions

1. Tenir enregistrat un nova tarifa (Preu/Hora).

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Tipus Preu".
2. El sistema comprova els tipus preu que hi ha a la base de dades que hi ha i els mostra en una taula i també mostra el formulari de introducció de dades noves, on demana el nom/descripció de la nova tarifa i el preu/hora en euros.
3. L'usuari introdueix les noves dades de nom/descripció i el nou preu/hora.
4. El sistema valida les dades.
5. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 4.1. El sistema detecta que les dades son errònies.
 - 4.1.1. Retorn al pas 3.
- 4.2. El sistema detecta que la tarifa ja existeix.
 - 4.2.1. Retorn al pas 3.

Cas d'ús "Modificar Tarifa Preus"

Actor:Usuari.

Pre-Condicions

1. Ha d'existir una tarifa de preus a la base de dades.

Post-Condicions

1. La tarifa de preus seleccionada amb les noves dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Modificar Tarifa Preus"
2. El sistema comprova les tarifes que hi ha a la base de dades que hi ha i els mostra en una taula.
3. L'usuari selecciona la tarifa que desitja modificar.
4. El sistema introdueix les dades de nom/descripció i preu/hora de la tarifa seleccionada al formulari.
5. L'usuari modifica el preu/hora al formulari i accepta.
6. El sistema valida les dades.
7. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 6.1. El sistema detecta dades incorrecte.
 - 6.1.1. Retorn al pas 5.

Cas d'ús "Eliminar Tarifa Preus"

Actor:Usuari.

Pre-Condicions

1. Ha d'existir una tarifa de preus a la base de dades.

Post-Condicions

1. La tarifa de preus eliminada permanentment de la base de dades.

Flux Normal

1. L'usuari inicia el cas d'ús "Eliminar Tarifa Preus"
2. El sistema comprova les tarifes que hi ha a la base de dades que hi ha i els mostra en una taula.
3. L'usuari selecciona la tarifa que desitja eliminar.
4. El sistema comprova que cap client l'estigui utilitzant i procedeix a l'acció.
5. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 6.2. El sistema detecta que hi ha un client que utilitza la tarifa seleccionada .
 - 6.2.1. Retorn al pas 3.

4.4.10. Gestió Països

Cas d'ús "Alta País"

Actor: Usuari.

Pre-Condicions

Post-Condicions

Tenir a la base de dades la informació d'un país.

Flux Normal

1. L'Usuari inicia el cas d'ús "Alta país".
 2. El sistema mostra els països existents en una taula i el formulari d'inserció de dades, identificador País i nom País.
-

3. L'usuari introdueix les dades al formulari i accepta.
4. El sistema valida les dades.
5. Finalitza el cas d'ús.

Flux Alternatiu

- 4.1. El sistema detecta que hi ha dades incorrectes.
 - 4.1.1. Retorn al pas 3.
- 4.1. El sistema detecta dades repetides
 - 4.1.1. Retorna al pas 3.

Cas d'ús "Modificar País"

Actor: Usuari.

Pre-Condicions

1. Tenir un país inserit a la base de dades.

Post-Condicions

1. Tenir a la base de dades la informació d'un país.

Flux Normal

1. L'Usuari inicia el cas d'ús "Modificació País".
 2. El sistema mostra els països existents en una taula i el formulari d'inserció de dades, identificador País i nom País.
 3. L'usuari selecciona el país que vol modificar.
 4. El sistema col·loca les dades del país seleccionat al formulari.
 5. L'usuari modifica les dades del formulari i accepta.
 6. El sistema valida les dades.
 7. Finalitza el cas d'ús.
-

Flux Alternatiu

6.1. El sistema detecta que hi ha dades incorrectes.

6.1.1. Retorn al pas 3.

6.1. El sistema detecta dades repetides

6.1.1. Retorna al pas 3.

4.4.11. Gestió Províncies

Cas d'ús "Alta Província"

Actor: Usuari.

Pre-Condicions

1. Tenir un país registrat a la base de dades.

Post-Condicions

1. Tenir a la base de dades la informació d'una província que pertanyi al país seleccionat.

Flux Normal

1. L'Usuari inicia el cas d'ús "Alta Província".
 2. El sistema mostra els països existents en una taula.
 3. L'usuari selecciona el país en que vols afegir una província.
 4. El sistema mostra un llistat de les províncies existents en el país seleccionat i un formulari on demana el identificador de província i el nom de la província.
 5. L'usuari introdueix les dades al formulari i accepta.
 6. El sistema valida les dades.
-

7. Finalitza el cas d'ús.

Flux Alternatiu

6.1. El sistema detecta que hi ha dades incorrectes.

6.1.1. Retorn al pas 3.

1.1. El sistema detecta dades repetides

1.1.1. Retorna al pas 3.

Cas d'ús "Modificació Província"

Actor: Usuari.

Pre-Condicions

1. Tenir una província guardada a la base de dades.

Post-Condicions

1. Tenir a la base de dades la nova informació de la província seleccionada.

Flux Normal

1. L'Usuari inicia el cas d'ús "Modificar Província".
 2. El sistema mostra els països existents en una taula.
 3. L'usuari selecciona el país en que desitja modificar una província.
 4. El sistema mostra un llistat de les províncies existents en el país seleccionat i un formulari on demana el identificador de província i el nom de la província.
 5. L'usuari selecciona una província de la llista.
 6. El sistema transfereix les dades de la província seleccionada al formulari.
 7. L'usuari modifica les dades al formulari i accepta.
 8. El sistema valida les dades.
 9. Finalitza el cas d'ús.
-

Flux Alternatiu

6.2. El sistema detecta que hi ha dades incorrectes.

6.2.1. Retorn al pas 3.

1.2. El sistema detecta dades repetides

1.2.1. Retorna al pas 3.

4.4.12. Gestió Poblacions

Cas d'ús "Alta Població"

Actor: Usuari.

Pre-Condicions

1. Tenir una província registrada a la base de dades.

Post-Condicions

1. Tenir a la base de dades la informació d'una població que pertanyi a la província seleccionada.

Flux Normal

1. L'Usuari inicia el cas d'ús "Alta Població".
 2. El sistema mostra els països existents en una taula.
 3. L'usuari selecciona el país en hi ha la província on es vol afegir la població.
 4. El sistema mostra un llistat de les províncies existents en el país seleccionat.
 5. L'usuari selecciona la província on vol afegir la nova població.
-

6. El sistema mostra un llista amb les poblacions existents de la província seleccionada, també un formulari on demana el identificador de província, l'identificador de població i el nom de la població.
7. L'usuari introdueix les dades al formulari i accepta.
8. El sistema valida les dades.
9. Finalitza el cas d'ús.

Flux Alternatiu

- 8.1. El sistema detecta que hi ha dades incorrectes.
 - 8.1.1. Retorn al pas 6.
- 8.2. El sistema detecta dades repetides
 - 8.2.1. Retorna al pas 3.

Cas d'ús "Modificar Població"

Actor: Usuari.

Pre-Condicions

1. Tenir alguna població registrada a la base de dades.

Post-Condicions

1. Tenir a la base de dades la informació modificada de la població seleccionada.

Flux Normal

1. L'Usuari inicia el cas d'ús "Modificació Població".
 2. El sistema mostra els països existents en una taula.
 3. L'usuari selecciona el país en hi ha la província del poble que es vol modificar.
 4. El sistema mostra un llistat de les províncies existents en el país seleccionat.
 5. L'usuari selecciona la província del poble que vol modificar.
-

6. El sistema mostra un llista amb les poblacions existents de la província seleccionada.
7. L'usuari selecciona la població que desitja modificar
8. El sistema col·loca les dades de la població seleccionada al formulari.
9. L'usuari modifica la dada nom poble.
10. El sistema valida les dades.
11. Finalitza el cas d'ús.

Flux Alternatiu

- 10.1. El sistema detecta que hi ha dades incorrectes.
 - 10.1.1. Retorn al pas 6.
- 10.2. El sistema detecta dades repetides
 - 10.2.1. Retorna al pas 3.

4.4.13. Gestió Codis Postals

Cas d'ús "Alta Codi Postal"

Actor: Usuari.

Pre-Condicions

1. Tenir una població registrada a la base de dades.

Post-Condicions

1. Tenir a la base el codi postal d'una població de la població seleccionada.

Flux Normal

1. L'Usuari inicia el cas d'ús "Modificar Codi Postal".
 2. El sistema mostra els països existents en una taula.
-

3. L'usuari selecciona el país on hi ha la província, de on és el poble que se li vol modificar el codi Postal.
4. El sistema mostra un llistat de les províncies existents en el país seleccionat.
5. L'usuari selecciona la província de on és el poble que se li vol afegir un codi Postal.
6. El sistema mostra un llista amb les poblacions existents de la província seleccionada.
7. L'usuari selecciona la població on hi vol afegir un codi postal.
8. El sistema mostra els codis postals existents del poble seleccionat i un formulari on demana el codi postal a inserir.
9. L'usuari introdueix les dades al formulari i accepta.
10. El sistema valida les dades.
11. Finalitza el cas d'ús.

Flux Alternatiu

10.1. El sistema detecta que hi ha dades incorrectes.

10.1.1. Retorn al pas 8.

10.2. El sistema detecta dades repetides

10.2.1. Retorna al pas 8.

Cas d'ús "Modificació Codi Postal"

Actor: Usuari.

Pre-Condicions

1. Tenir un codi postal registrat a la base de dades.

Post-Condicions

1. Tenir a la base el codi postal seleccionat amb les noves dades introduïdes.
-

Flux Normal

1. L'usuari inicia el cas d'ús "Modificació Codi Postal".
2. El sistema mostra els països existents en una taula.
3. L'usuari selecciona el país on hi ha la província, de on és el poble que se li vol modificar un codi Postal.
4. El sistema mostra un llistat de les províncies existents en el país seleccionat.
5. L'usuari selecciona la província de on és el poble que se li vol modificar el codi Postal.
6. El sistema mostra un llista amb les poblacions existents de la província seleccionada.
7. L'usuari selecciona la població on hi vol modificar codi postal.
8. El sistema mostra els codis postals existents del poble seleccionat
9. L'usuari selecciona el codi postal que desitja modificar.
10. El sistema transfereix les dades del codi postal seleccionat al formulari per modificar el codi postal.
11. L'usuari modifica les dades al formulari i accepta.
12. El sistema valida les dades.
13. Finalitza el cas d'ús.

Flux Alternatiu

- 12.1. El sistema detecta que hi ha dades incorrectes.
 - 12.1.1. Retorn al pas 8.
- 12.2. El sistema detecta dades repetides
 - 12.2.1. Retorna al pas 8.

4.4.14. Gestió Operacions

Cas d'ús "Alta Operació"

Actor: Usuari.

Pre-Condicions

Post-Condicions

1. Tenir emmagatzemada a la base dades una operació.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Operació".
2. El sistema mostra tots els tipus operacions existents i el formulari per introduir nous tipus operacions, on demana el nom de l'operació.
3. L'usuari introdueix el nom de la nova operació.
4. El sistema valida les dades.
5. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 4.1. El sistema detecta dades incorrectes o que l'operació esta repetida.
 - 4.1.1. Retorn al pas 3.

Cas d'ús "Modificar Operació"

Actor: Usuari.

Pre-Condicions

1. Que existeixi una operació a la base de dades.

Post-Condicions

1. Tenir emmagatzemada a la base dades l'operació.
-

Flux Normal

1. L'usuari inicia el cas d'ús "Modificar Operació".
2. El sistema mostra tots els tipus operacions existents i el formulari per introduir nous tipus operacions, on demana el nom de l'operació.
3. L'usuari selecciona l'operació que vol modificar.
4. El sistema carrega al formulari les dades de l'operació seleccionada.
5. El sistema valida les dades.
6. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 5.1. El sistema detecta dades incorrectes o que l'operació esta repetida.
 - 5.1.1. Retorn al pas 4.

4.4.15. Gestió Material

Cas d'ús "Alta Material"

Actor: Usuari.

Pre-Condicions

Post-Condicions

1. Tenir emmagatzemada a la base dades un material.

Flux Normal

1. L'usuari inicia el cas d'ús "Alta Material".
 2. El sistema mostra tots els tipus de materials existents i el formulari per introduir nous tipus materials, on demana el nom del material.
 3. L'usuari introdueix el nom de nou material.
-

4. El sistema valida les dades.
5. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 4.2. El sistema detecta dades incorrectes o que el material ja està a la base de dades.
 - 4.2.1. Retorn al pas 3.

Cas d'ús “Modificar Material”

Actor: Usuari.

Pre-Condicion

1. Que existeixi un material a la base de dades.

Post-Condicion

1. Tenir emmagatzemada a la base de dades el material seleccionat amb les noves dades.

Flux Normal

1. L'usuari inicia el cas d'ús “Modificar Material”.
2. El sistema mostra tots els tipus de materials existents i el formulari per introduir nous materials, on demana el nom del material.
3. L'usuari selecciona el material que vol modificar.
4. El sistema carrega al formulari les dades del material seleccionat.
5. El sistema valida les dades.
6. El sistema finalitza el cas d'ús.

Flux Alternatiu

- 5.2. El sistema detecta dades incorrectes o que el material està repetit.
-

5.2.1. Retorn al pas 4.

4.5. Diagrama de Classes del Domini

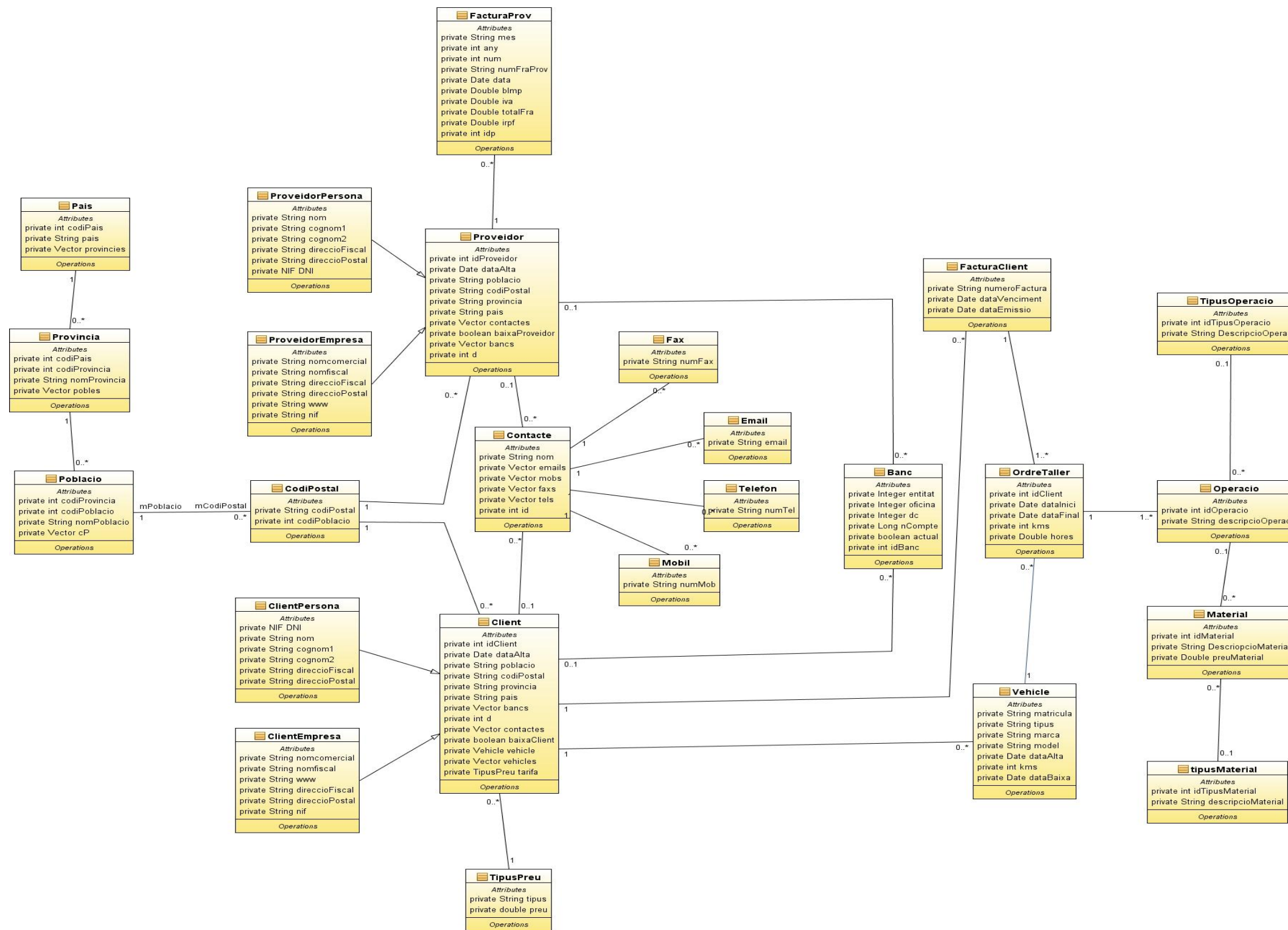


Figura 3. Diagrama de Classes del Domini

A la pàgina anterior es mostra el diagrama de classe del domini, és a dir tots els objectes que engloben la lògica de l'aplicació. De cada objecte, en aquest diagrama, s'ha representat les relacions que mantenen els objectes entre sí i els atributs o variables de classe que fa servir. S'han obviat les operacions ja que en totes les classes son els constructors amb tots els paràmetres, i tots els mètodes get i set de cada atribut.

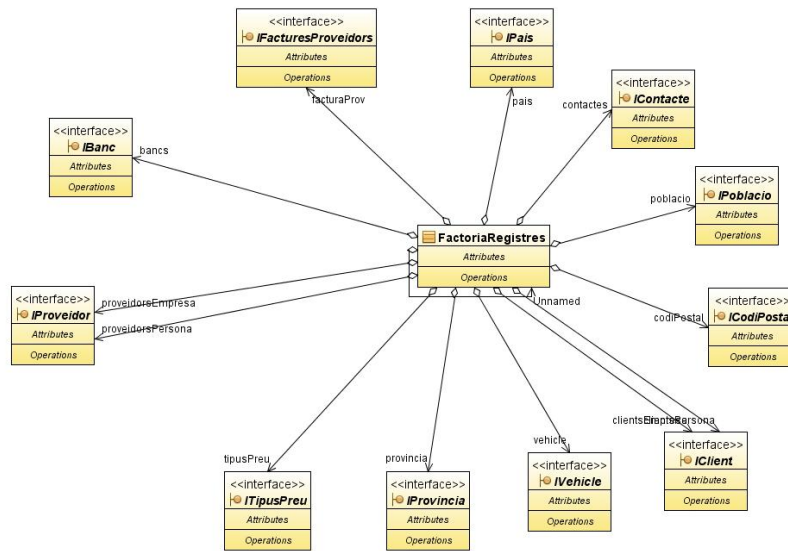


Figura 4. Interfícies i Factoria Registres

Les interfícies mostrades al damunt, són les interfícies que fan de pont entre les classes del domini i la classes de la persistència. Hi ha pràcticament una interfície per a cada classe del domini, així aconseguim aplicar el patró indirecció.

Durant la realització d'aquest diagrama es van tenir problemes de disseny ja que era complicat el tema de la generalització. El aquest cas s'ha optat per fer la generalització de client i proveïdor, perquè per a fer algunes operacions importants ens era d'utilitat poder diferenciar de forma ràpida entre clients i proveïdors, tot i que, en d'altres casos seria més d'utilitat tenir el la generalització persona i empresa.

Un altre punt fort i complicat del disseny és tot el tema d'ordres de taller, factures, operacions, materials, ja que no acaben de funcionar com a la majoria de tallers, per

solucionar aquesta diferència amb d'altres tallers, s'ha inclòs dues taules, "TipusOperacio" i "TipusMaterial".

4.6. Digrames de seqüència del sistema

4.6.1. Cas d'ús "Alta Ordre Taller"

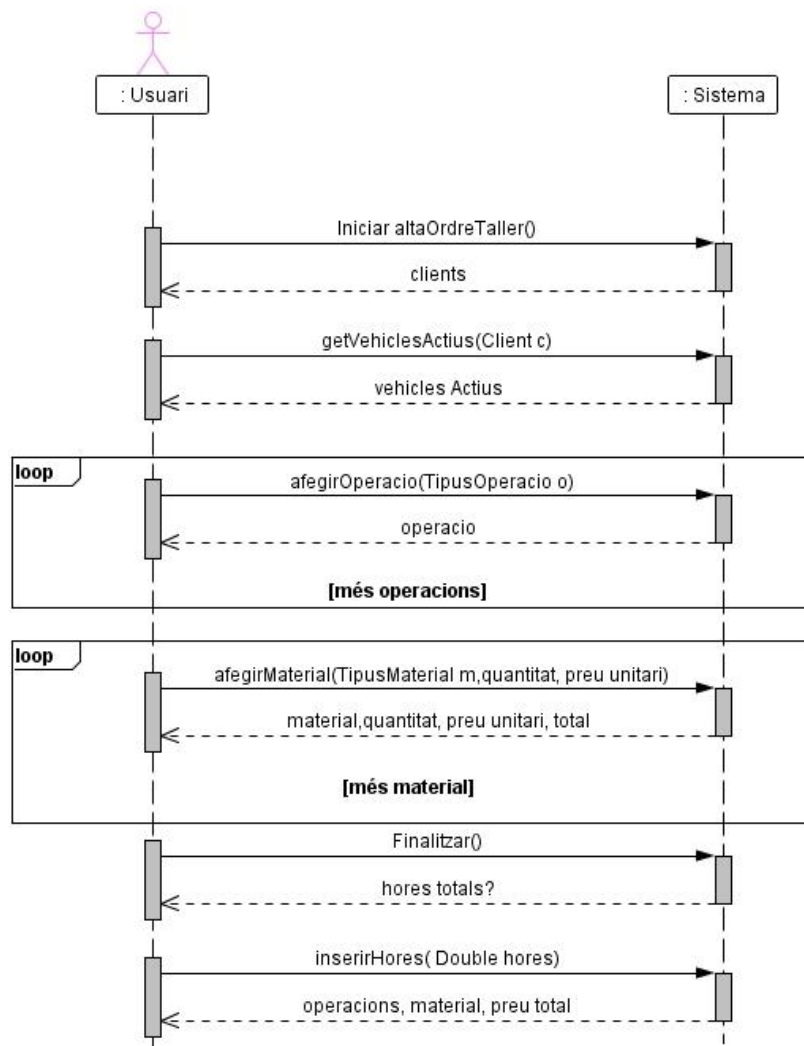


Figura 5. Diagrama de seqüència dels sistema (altaOrdreTaller)

4.6.2. Cas d'ús "Alta Factura Client"

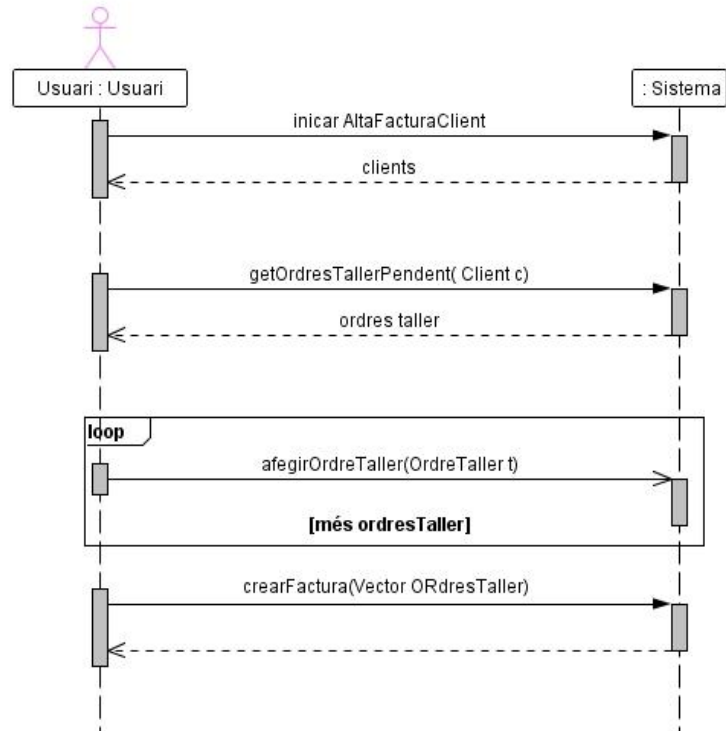


Figura 6. Diagrama de seqüència del sistema (altaFacturaClient)

5. Disseny

5.1. Patró arquitectònic

Per desenvolupar aquesta aplicació s'utilitza una arquitectura MVC, dividit en 4 capes, seguint la metodologia del patró "capes." La lògica d'aquest software és complexa és interessant salvaguardar-la i aïllar-la de la resta de l'aplicació, principalment de la presentació ja que tendeix a canviar. En el gràfic següent és mostra el funcionament de l'esmentada arquitectura.

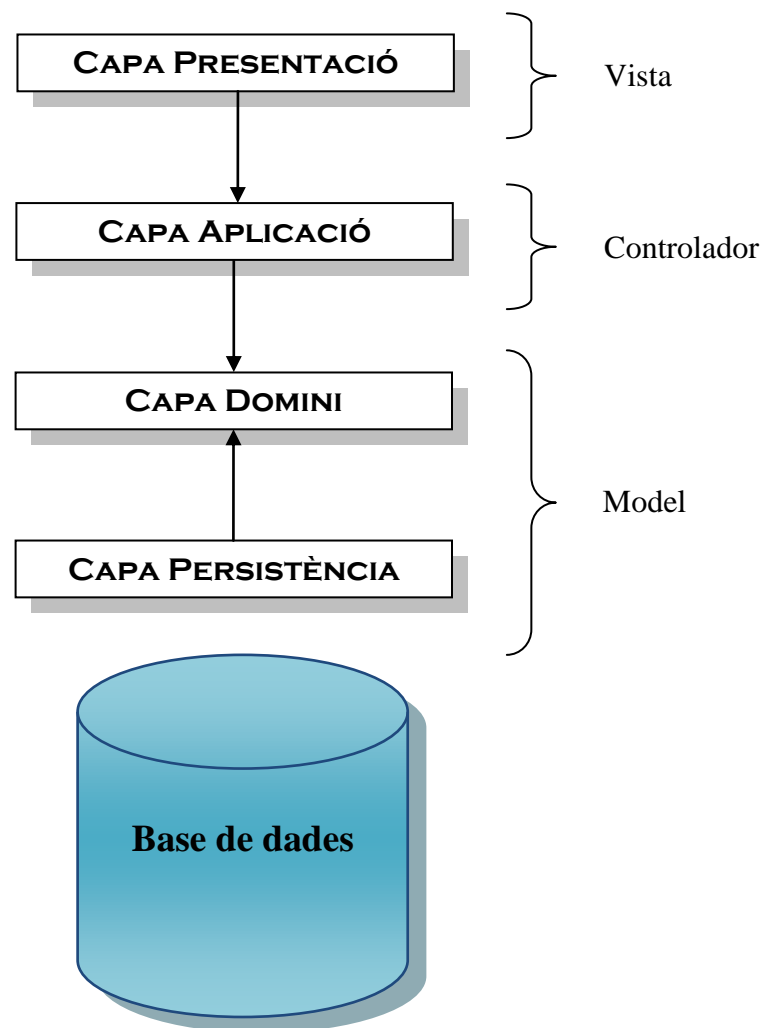


Figura 7. Esquema MVC 4 capes

L'arquitectura en capes permet separar la implementació en diferents funcionalitats, cada capa s'encarrega d'unes funcionalitats diferents. A grans trets s'explica que fa cada capa:

- **Capa Presentació :** És la que s'encarrega de la interacció amb l'usuari, és a dir, és des de on l'usuari es comunica amb el sistema i viceversa.
- **Capa Aplicació:** S'encarrega de transformar les dades de la *presentació* en objectes del *domini*, i les transfereix cap a la capa Domini. També fa la funció inversa, rep els objectes del domini i els prepara per a que la capa Presentació, en pugui utilitzar la informació dels objectes.
- **Capa Domini:** És la capa que conté els objectes del domini i la que es responsabilitza de comunicar les peticions de la capa Aplicació cap a la capa Persistència. Al inrevés, retorna els resultats del capa Persistència cap a la capa Aplicació.
- **Capa Persistència:** Aquesta capa s'encarrega de realitzar les ordres de consulta, modificació i inserció de la capa superior, en el llenguatge que usa la Base de Dades, el cas d'aquest projecte, fa la conversió de java a Sql.

5.1.1. Capa Presentació

Com s'ha explicat anteriorment, és la que s'encarrega de la interacció amb l'usuari, rep i les ordres de l'usuari i les hi retorna.

En la presentació s'hi troben totes les classes que són pantalles, és a dir, classes del tipus JFrame, JInternalFrame, JPanel.

També hi ha unes classes que utilitzen els formularis de les finestres, per tal de reduir el nombre de comprovacions en el moment de confirmar l'acció. Alguns exemples, d'aquestes classes son:

La classe “LimitadorCaractersMax”: el que fa es limitar el nombre de caràcters que es poden introduir en un camp d’un formulari.

La classe “LimitadorNomesLLetres”: té la mateixa funcionalitat que l’anterior, és a dir, limita el nombre màxim de caràcters que hi podem escriure en camp, però a més a més, restringeix que aquets caràcters no poden ser números.

La classe “LimitadorNomesNum”: Com indica el nom, només ens hi deixa escriure un nombre limitat de caràcters compresos entre el 0 i el 9.

5.1.2. Capa Aplicació

Aquesta capa equival al controlador del Patró MVC. Aquest patró assigna la responsabilitat de rebre o tractar un esdeveniment del sistema cap a una subsistema determinat. Un error molt comú és sobrecarregar el controlador amb tasques que no li corresponen.

En aquesta capa hi trobem el següents controladors:

- **Controlador Gestió Clients:** dins d’aquesta classe s’engloben tots els mètodes que necessiten per fer qualsevol operació amb els clients del taller, des de donar d’alta, modificar, llistar clients, etc...
 - **Controlador Gestió Proveïdors:** De la mateixa manera que el controlador anterior conté tots els mètodes que fan referència als proveïdors.
 - **Controlador Gestió Contactes:** Aquest controlador posseeix totes les funcions que necessiten els contactes dels clients i proveïdors.
 - **Controlador Gestió Bancs:** Com ens els casos anteriors aquest controlador té tots els mètodes referents a la gestió de bancs.
-

- **Controlador Gestió Administració:** Conté mètodes que no son específics de cap objecte, sinó que son genèrics de tot el software, com exemple, les altes de poblacions, províncies i països, llistar les poblacions, províncies, establir el preu/hora, etc..
- **Controlador Factures Proveïdors:** És el controlador que posseeix les funcions que engloben tot el que fa esment a les factures dels proveïdors, altes, baixes, modificacions, etc...

5.1.3. Capa Domini

És la capa on hi ha tots els objectes, i és l'encarregada de comunicar els esdeveniments que ha sol·licitat la capa Aplicació. En el domini hi resideix tota la lògica de l'aplicació. Un objecte del domini mai accedirà a la base de dades directament sinó que ho farà mitjançant una interfície. Per evitar acoblament s'ha fet servir el patró Factoria.

Algunes de les classes del domini que hi ha són:

- ClientEmpresa
- ClientPersona
- ProveïdorEmpresa
- ProveïdorPersona
- Vehicle
- Banc
- Contacte
- IClient
- IProveïdor
- IVehicle
- IBanc
- IContacte
- TipusPreu
- FactoriaRegistres

5.1.4. Capa Persistència

La capa Persistència equival també a l'entorn del Model del MVC. A la persistència s'hi troben tots els mètodes que interaccionen amb la base de dades, aquests mètodes contenen sentències SQL.

En aquesta capa es fan servir classes mirall, és a dir, hi ha totes les classes que hi ha el domini però ara la seva feina és la comunicar-se amb la base de dades. Aquestes classes mirall es defineixen amb una implementació de la interfícies que els hi pertoquen.

Una de les classes més importants de la persistència és “ConnexioBD”, que s’encarrega de crear la connexió amb la base de dades corresponent mitjançant el JDBC. Cada classe de la persistència al ser creada crida una instància de “ConnexioBD” . Per tal de que el software sigui més eficient la connexió amb la base de dades es crea al principi i no es tanca fins que es surt de l’aplicació, ja que aquesta aplicació no te el problema de la concurrència.

5.2.Patrons de Disseny

5.2.1. Singleton

És un patró molt simple que ajuda a l’aplicació a no sobre carregar-se innecessariament amb objectes instanciats varies vegades. Es fa servir quan en temps d’execució només és necessita una sola instància d’un objecte determinat. Aquest patró s’ha fet servir per exemple en la creació de les finestres de la capa presentació, a la classe “ConnexioBD” i a la classe “FactoriaRegistres”.

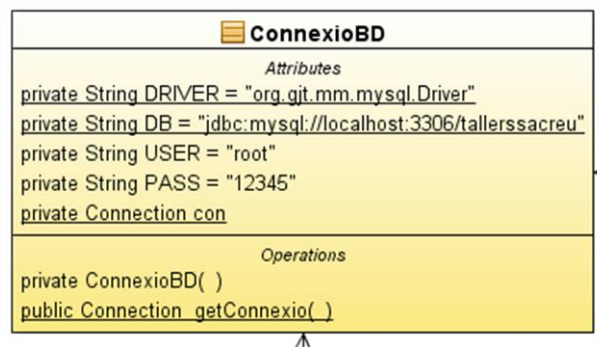


Figura 8: Patró Singleton a la classe ConnexioBD

5.2.2. Factoria

L'objectiu d'aquest patró és separar la part del codi de creació d'objectes de la resta d'aplicació. Agafa sentit quan es creen objectes a partir d'interfícies, on la lògica és més complexa. El patró factoria s'ha fet servir en la classe "FactoriaRegistres".

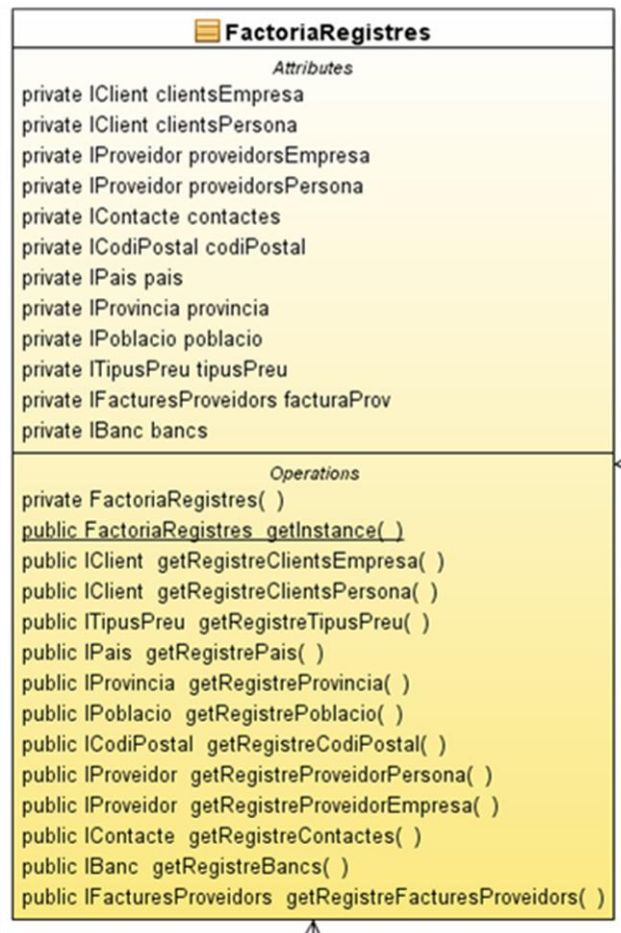


Figura 9: Patró Factoria a FactoriaRegistres

5.2.3. Indirecció

Amb aquest patró es millora el baix acoblament entre dues classes assignant responsabilitats de la mediació entre ells a un tercer. Per exemple, entre la classe "Vehicle" del domini i la classe "VehicleBD" de la persistència existeix una interfície "IVehicle" que assigna les responsabilitats de la intervenció entre elles i igual que amb

aquestes dues classes, el patró indirecció s'aplica a totes els capes de domini que estan relacionades amb la persistència.

5.2.4. Controlador

El patró controlador és un patró que serveix com a intermediari entre una determinada interfície i l'algorisme que la implementa, de tal manera que és la rep les dades de l'usuari i les envia a les diferents classes segons el mètode cridat. Aquest patró suggereix que la lògica de negocis ha d'estar separada de la capa de presentació, això serveix per a augmentar la reutilització de codi i alhora tenir un control superior. Es recomana dividir els esdeveniments del sistema en el màxim nombre de controladors per a poder augmentar la cohesió i disminuir l'acoblament.

5.3. Base de dades

En les dues següents pàgines es mostra el model conceptual i el model relacional de la Base de Dades.

5.3.2. Model relacional

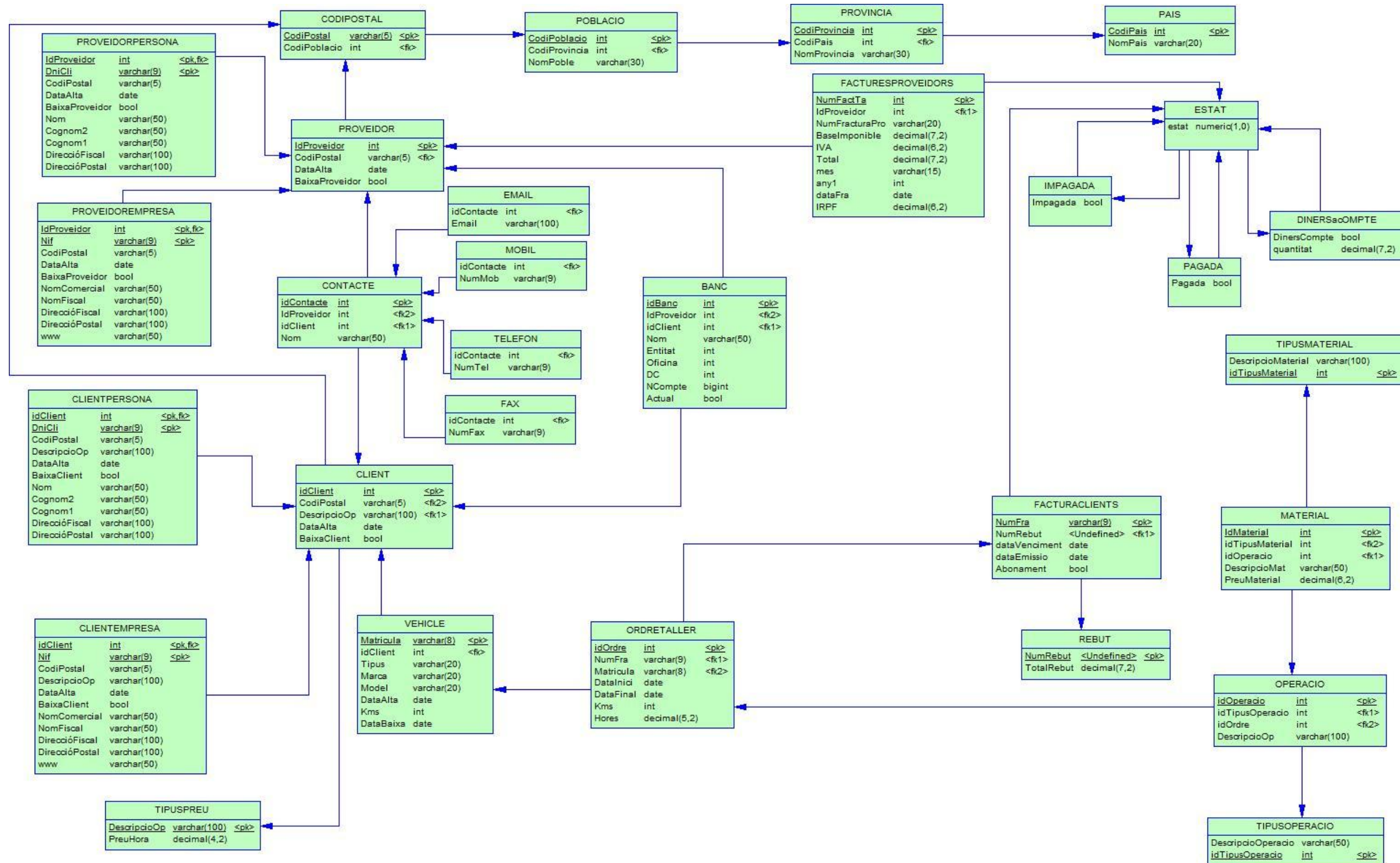


Figura 11. Model Relacional de la Base de Dades

5.3.3. Regles de negoci

Taula “Proveïdor”

- El valor de la columna “dataAlta” ha de ser igual o anterior a la data actual.

Taula “Client”

- El valor de la columna “dataAlta” ha de ser igual o anterior a la data actual

Taula “TipusPreu”

- El valor de la columna “preuHora” ha de major o igual a 0.

Taula “Vehicle”

- El valor de la columna “dataAlta” ha de ser anterior o igual al valor de la columna “dataBaixa”.

Taula “OrdreTaller”

- El valor de la columna “dataInici” ha de ser anterior o igual al valor de la columna “dataFinal”.
- El valor de la columna “Hores” ha de ser igual o superior a 0.

Taula “FacturaClient”

- El valor de la columna “dataEmissió” ha de ser anterior o igual al valor de la columna “dataVenciment”.
-

Taula “FacturesProveidor”

- El valor de la columna “mes” ha de ser “Gener”, “Febrer”, “Març”, “Abril”, “Maig”, “Juny”, “Juliol”, “Agost”, “Setembre”, “Octubre”, “Novembre” o “Desembre”.
- El valor de la columna “any1” ha de ser anterior o igual al any actual.
- El valor del mes de la columna “dataFra” ha de ser anterior o igual al valor de la columna “mes”.

Taula “Material”

- El valor de la columna “preuMaterial” ha de ser igual o superior a 0.

5.3.4. Seqüències

La majoria de taules utilitzen un identificador numèric, però no tots els ha d’inserir l’usuari sinó que ho fa automàticament la base de dades. El cas d’aquest projecte és interessant que les seqüències comencin a 1, i incrementin de un en un. Aquest problema el resol el mySQL amb una funció predefinida anomenada “AUTO_INCREMENT”. La funció en si per defecte comença per 1 i incrementa de forma unitària.

Relació de taules on es fa servir la funció “Auto_Increment” :

- Contacte
- Banc
- Material
- TipusMaterial
- Operació
- TipusOperació

La forma de indicar que un atribut utilitza aquesta funció, és la creació de la taula, en l’exemple que hi ha a continuació es veu més clar.

```

CREATE TABLE BANC
(
  IDBANC          INT          NOT NULL  AUTO_INCREMENT,
  NOM             VARCHAR(50),
  ENTITAT        INT          NOT NULL,
  OFICINA        INT          NOT NULL,
  DC              INT          NOT NULL,
  NCOMPTE        BIGINT       NOT NULL,
  IDCLIENT       INT          NOT NULL,
  ACTUAL         TINYINT(1)   NOT NULL,
  PRIMARY KEY (IDBANC)
);

```

5.4. Diagrama de seqüència del cas d'ús "AltaOrdreTaller"

Primer de tot es recuperen les dades del client amb els vehicles, i contactes si en té.

Per tant el diagrama d'aquest pas és:

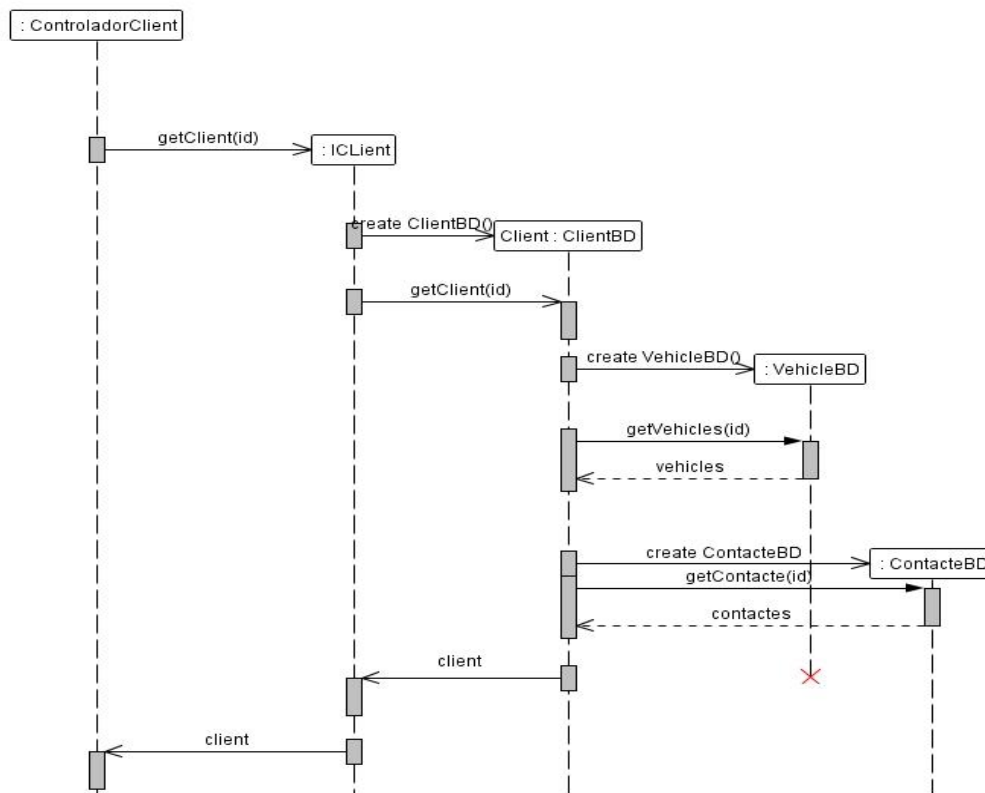


Figura 12. Diagrama de seqüència AltaVehicle 1

El següent pas és la introducció de dades d'ordre taller, i això ho representa el diagrama següent:

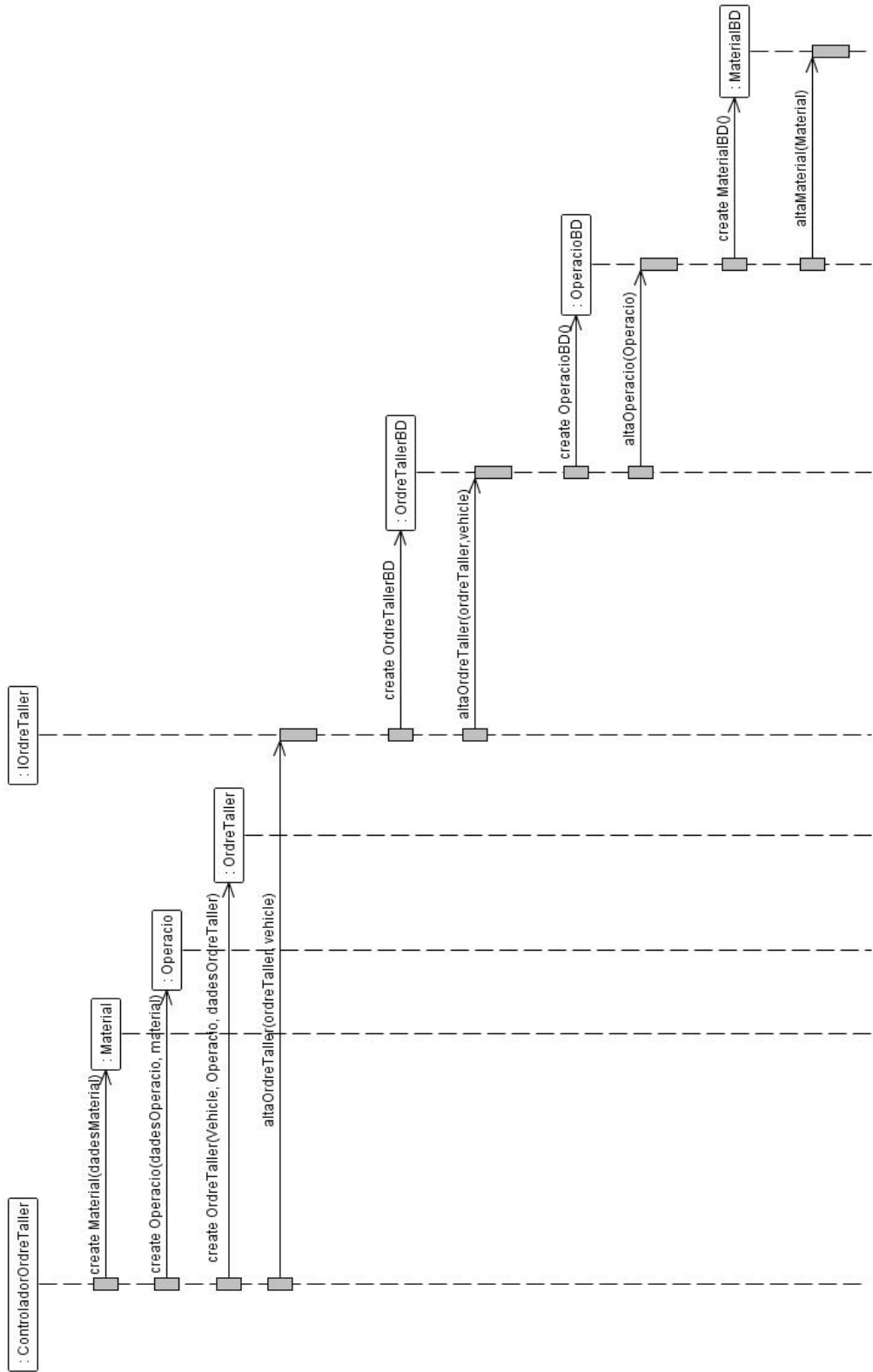


Figura 13. Diagrama altaOrdreTaller 2

6. Implementació

6.1. Interfície gràfica

És important recalcar que per optimitzar el software, es treballa en memòria i en base de dades, és a dir, quan es vol recuperar informació primer es va al magatzem de dades, i si no es troba la informació demanada llavors es fa la consulta a la base de dades.

En l'apartat d'objectius s'ha vist que una fita important és que la interfícies sigui fàcil de fer servir i per tant la interfície és molt guiada. El que s'ha fet per evitar errors prematurs per equivocació alhora de prémer tecles i aconseguir que la interfície sigui usable, ràpida, fluida, i alhora robusta, és el següent:

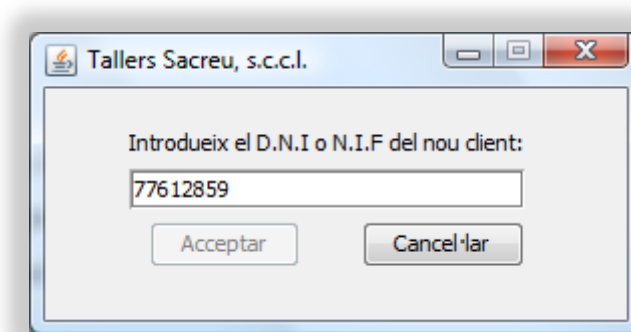


Figura 14. Finestra seleccióAlta

Si es mira la imatge del JFrame, on s'introdueix el dni o nif, el botó "Acceptar" està desactivat fins que no hi ha inserits 9 caràcters, i apart s'utilitza la classe "LimitadorCaractersMax", aquesta classe només controla la longitud màxima del camp. En aquest cas com que dni, té vuit nombres i una lletra es fa servir la classe LimitadorCaractersMax, que deixar inserir qualsevol tipus de caràcter.

```
public class LimitadorCaractersMax extends PlainDocument {
    private JTextField miJTextField;
    private int nroMaxCaracteres;
    public LimitadorCaractersMax(JTextField mijtext, int nroMaxCaract){
        miJTextField=mijtext;
        nroMaxCaracteres=nroMaxCaract;
    }
    public void insertString(int arg0, String arg1, AttributeSet arg2)
throws BadLocationException
    {
        for (int i=0;i<arg1.length();i++)
            if
((miJTextField.getText().length()+arg1.length())>nroMaxCaracteres)
                return;
        super.insertString(arg0, arg1, arg2);
    }
}
```

Com s'ha comentat en el capítol de disseny i més concretament en l'apartat de la capa Presentació, es fan servir unes classes per controlar l'entrada de dades. Amb això s'aconsegueix més velocitat i fluïdesa alhora de dur a terme la introducció de dades.

Hi ha altres classes que limiten que només es puguin introduir una certa quantitat de nombres i una altra que només una certa quantitat de lletres. El paquet Swing de Java, té un component d'entrada anomenat JFormattedText, que no deixa de ser un JTextField on hi ha límits que hi vols que s'hi escrigui i en quin format.

En quan a la introducció de dates s'ha de fer servir un calendari així es limita el format de la data i la presentació té menys feina de comprovar que allò és una data. El calendari que s'utilitza és un JCalendar, extret de www.toedter.com i modificat específicament per a aquesta aplicació.

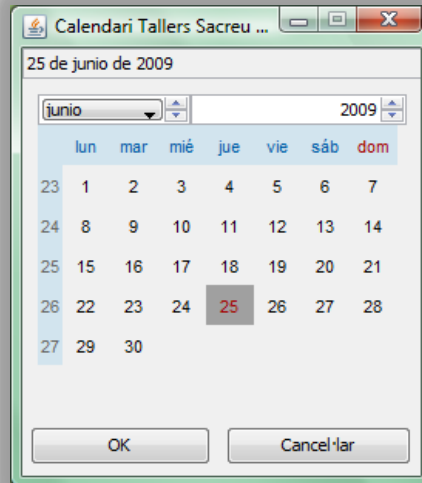


Figura 15. JCalendar de Toedter

Una altre detall important de la implementació, es que cada cop l'usuari desitja fer una modificació o una alta d'algun client, banc, vehicle, primer se li mostren totes les dades del client per pantalla i així pot veure abans d'introduir les dades si ja disposa d'elles. Anem a veure un exemple de l'alta Vehicle.

Primer es seleccionaria quin a quin client se li vos afegir un vehicle.

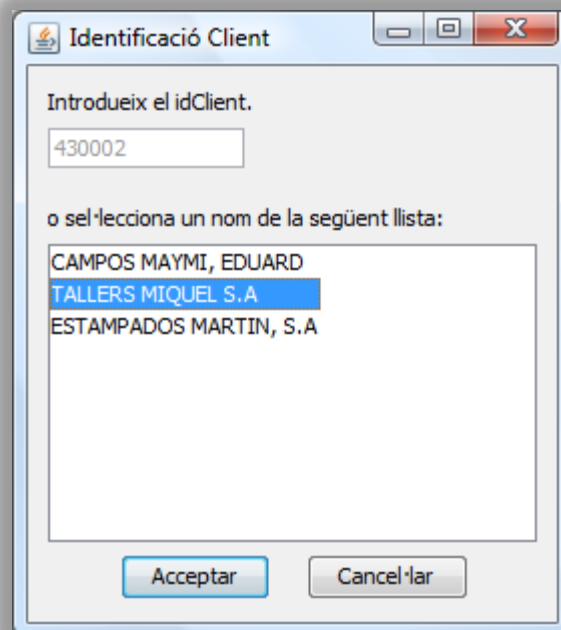


Figura 16. Finestra IDClient

Un cop acceptem ens mostra la finestra de la següent pantalla.

On s'hi pot veure les dades del client, les dades dels contactes dels que disposa, una taula amb els seus vehicles i finalment un formulari a on inserir les dades del nou vehicle.

Alta de Vehicles

Dades

idClient: 430002
 Nom: TALLERS MIQUEL S.A
 DNI: B12345678
 Direcció Fiscal: C/ OLLERIA, 23
 Població: Arenys de Mar
 Contacte: TALLERS EDU

Telefons	T.Mòbils	Faxes	Emails
	649753696		

Dades del Vehicle

Tipus: Altres
 Data Alta:
 Marca:
 Model:
 Matricula:
 Kms:

Vehicls Existents

Matricula	Tipus	Marca	Model	DataAlta	De Baixa?
1234HYT	CAMIÓ	IVECO	SAAS	2009-06-25	<input type="checkbox"/>
1280FWX	ALTRES	CITROEN	BERLINGO	2009-06-18	<input type="checkbox"/>
6320CRD	ALTRES	RENAULT	CLIO	2009-06-18	<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>
					<input type="checkbox"/>

Acceptar Cancel.lar

Figura 17. Finestra altaVehicle

6.2. Controladors

Per seguir el patró controlador, s'ha implementat un controlador, per a cada subsistema, és a dir, hi ha el Controlador de Clients, que inclou totes les operacions relacionades amb els clients, Controlador Gestió Factures de Proveïdors que té implementades totes els mètodes necessaris per que funcioni aquest subsistema de l'aplicació. Els cada cop que és crida a un controlador es crida a una instancia ja que utilitza el patró Singleton.

6.3. Interfícies i Factoria de Registres

En l'apartat de disseny s'ha comentat que un objecte del domini no pot accedir a la base de dades directament sinó que ho fa mitjançant una interfície. Com que les interfícies no tenen constructor, no es pot fer un “new” de la interfície i la persistència queda fora de l'abast de l'aplicació per fer un nou objecte d'una classe de la persistència.

A continuació es mostra com s'ha solucionat el problema esmentat.

```
public IVehicle getRegistreVehicles() {
    try {
    if (vehicle == null) {
    vehicle = (IVehicle) Class.forName("Persistencia.VehicleBD").newInstance();
    }
    } catch (ClassNotFoundException e) {
        System.out.println("Classe no trobada: " + e.getMessage());
    } catch (InstantiationException e) {
        System.out.println("Instanciació fallida: " + e.getMessage());
    } catch (IllegalAccessException e) {
        System.out.println("Acces ilegal: " + e.getMessage());
    } catch (Exception e) {
        System.out.println("Exception: " + e.getMessage());
    }
    return vehicle;
}
```

Utilitzant el “Class.forName(“Persistencia.VehicleBD”).newInstance();”, el que s’aconsegueix fer un recuperar una instància de la classe “VehicleBD”. El codi mostrat és un mètode de la classe “FactoriaRegistres”.

6.4. Persistència

En la implementació de la persistència s’han creat classes mirall, és a dir, hi ha una classe per cada taula de la base de dades. Aquí s’implementen totes mètodes de per accedir a la base de dades. Per seguretat en les insercions es comprova que tot el procés ha funcionat correctament ja que sinó no fa el “commit” i la execució fa un “rollback”. Per exemple, en el codi de sota es veu com es dona d’alta un contacte.

En el constructor de la classe ja fa la connexió.

```

public ContacteBD() throws ExcepcioDAOFallaConnexio {
    try {
        con = ConnexioBD.getConnexio();
    } catch (Exception exec) {
        throw new ExcepcioDAOFallaConnexio("Falla la connexió", exec);}
}

```

Després ja es fa el mètode alta contacte:

```

public void altaContacteBD(int ids, Contacte contacte) throws Exception {
    con = ConnexioBD.getConnexio();
    PreparedStatement stm = null;
    boolean committed = false;
    String sql;
    int idContacte;
    int idProv=-1;
    int idCli=-1;
    String id=String.valueOf(ids);
    if(id.startsWith("43")){
        idCli=ids;
    }
    else if(id.startsWith("40")||id.startsWith("41")){
        idProv=ids;
    }
    try {
        con = ConnexioBD.getConnexio();
        sql = "INSERT INTO CONTACTE (NOM,IDPROVEIDOR,IDCLIENT) VALUES
(?) , (? , ?) ";
        stm = con.prepareStatement(sql);
        stm.clearParameters();
        Contacte c = contacte;
        String nom = c.getNom();
        stm.setString(1, nom);
        stm.setInt(2, idProv);
        stm.setInt(3, idCli);
        stm.executeUpdate();
        idContacte = this.getLastId();
        System.out.println(idContacte);
        TelefonBD tBD = new TelefonBD();
        if (!c.getTels().isEmpty()) {
            tBD.AltaTelefon(idContacte, c.getTels());
        }
        FaxBD fBD = new FaxBD();
        if (!c.getMobs().isEmpty()) {
            fBD.altaFaxBD(idContacte, c.getFaxes());
        }
        MobilBD mBD = new MobilBD();
        if (!c.getMobs().isEmpty()) {

```



```
        mBD.altaMobilBD(idContacte, c.getMobs());
    }
    EmailBD eBD = new EmailBD();
    if (!c.getEmail().isEmpty()) {
        eBD.altaEmailBD(idContacte, c.getEmail());
    }
    con.commit();
    committed = true;
} catch (SQLException e) {
    if (e.getMessage().startsWith("ORA-00001")) {
        throw new ExcepcioDAOPKDuplicada("error", e);
    } else {
        throw new ExcepcioDAOFallaConnexio("Falla la connexió", e);
    }
} catch (Exception ex) {
    throw new ExcepcioDAO(ex.getMessage(), ex);
} finally {
    try {
        if (stm != null) {
            stm.close();
        }
        if (!committed) {
            con.rollback();
        }
    } catch (SQLException e) {
        throw new ExcepcioDAOFallaConnexio("Falla la connexió", e);
    }
}
}
```


7. Conclusions

L'aplicació que es presenta en aquest projecte final de carrera està pensada, dissenyada i elaborada per a un client en concret. Amb això es vol dir que altres empreses del sector segurament no la puguin utilitzar ja que s'ha desenvolupat en base a els processos concrets de Tallers Sacreu s.c.c.l.

D'altra banda es proposa una metodologia software pel desenvolupament de petits i mitjans sistemes d'informació. El temps ha estat el principal factor pel qual no s'ha pogut seguir fil per randa metodologies tan conegudes com Unified Process o Mètrica v3. Ja que estan pensades per a projectes molt més grans que l'aplicació que s'ha desenvolupat en aquest projecte. A causa del factor temporal també hi ha algun aspectes com algunes proves que s'han quedat en segon terme.

A diferència de la majoria de projectes finals de carrera aquest té al darrera un client real. Això ha permès aconseguir una experiència molt més fructífera i gratificant, a causa de saber que el software que es dissenya realment tindrà un ús en la vida quotidiana del client. Això és l'autèntica satisfacció, saber que el que fas serveix d'alguna cosa o ajuda a les necessitats d'alguna persona.

Eduard Campos Maymi
Escola Universitària Politècnica de Mataró
Enginyeria Tècnica en Informàtica de Gestió
Primavera 2009

Gestió Tallers Sacreu s.c.c.l

cammayed@eupmt.upc.edu

Resum: *Tallers Sacreu S.C.C.L és una petita empresa de reparació de vehicles que amb el pas del temps ha vist com el sistema de gestió del seu negoci ha quedat obsolet. Actualment funciona amb fulls de càlcul i processadors de text, i existeix una manca important d'ordre. L'objectiu del projecte és millorar la gestió, oferir a aquesta empresa l'agilització de tots els seus processos i evitar la pèrdua de dades mitjançant l'elaboració d'una aplicació que permetrà controlar de forma centralitzada tots les funcions i processos del taller, referents a la facturació.*

1. Introducció

Aquesta aplicació està dissenyada i desenvolupada seguint les necessitats d'un client real, Tallers Sacreu S.C.C.L.

Tallers Sacreu, S.C.C.L és una empresa dedicada des de fa més de 25 anys a la reparació general d'automòbils, especialment de vehicles industrials, Empresa d'àmbit local que s'ha consolidat amb una cartera de clients fixes d'arreu de la comarca del maresme, tant empreses com particulars. Estem parlant de més de 250 clients que al llarg de l'any passen a fer les revisions i reparacions al seus vehicles.

Pel que fa als proveïdors, actualment treballen amb més d'una cinquantena de proveïdors habituals i altres d'espòrics, Entre els proveïdors amb els que treballa s'hi troben proveïdors de serveis, proveïdors ordinaris, és a dir proveïdors de mercaderies, matèries primeres etc, i finalment amb Administracions Públiques.

La necessitat d'aquest software es deu a que el sistema d'informació del taller s'ha quedat obsolet. El problema del seu sistema de treball actual és la pèrdua d'informació i el poc control de gestió sobre clients, proveïdors i la facturació. L'objectiu de l'aplicació és resoldre aquest problema de la gestió i sobretot solucionar el problema de la pèrdua d'informació i agilitzar els processos de gestió de clients, proveïdors i factures.

2. Objectius

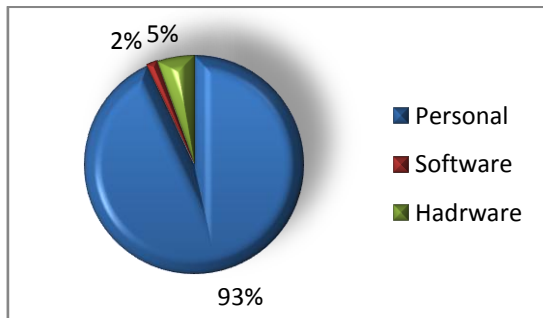
L' objectiu principal d'aquest projecte és desenvolupar una aplicació escalable, usable, és a dir, que sigui molt fàcil de d'utilitzar amb interfícies gràfiques intuïtives que no provoquin redundància sobre les dades que s'ha d'introduir, ha de ser mantenible i robusta am un bon control d'errors.

Final ha de resoldre la gestió dels clients, proveïdors i facturació, i sobre tot la problemàtica de la pèrdua de dades.

3. Pressupost

S'ha estimat que el desenvolupament del projecte tindrà una durada aproximada de 411 hores. El cost total de l'aplicació comptant amb el costos derivats de personal, de hardware i de llicències software ascendeix a la quantitat de 5.310 € distribuïts de la següent manera:

- 4.939 € de personal.
- 88 € de software.
- 283 € de hardware.

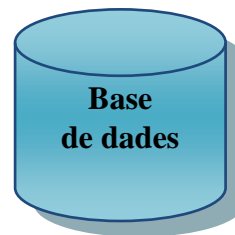
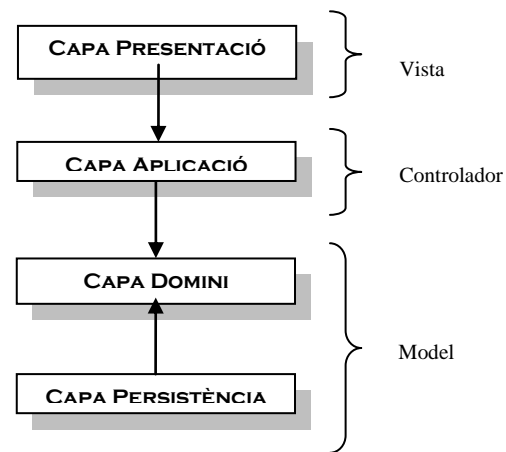


Amb el gràfic s'observa que la majoria de costos deriven de personal ja que les aplicacions utilitzades són lliures o versions d prova, i el hardware utilitzat és un portàtil de gamma mitja.

4. Disseny

Patró arquitectònic

Per desenvolupar aquesta aplicació s'utilitza una arquitectura MVC, dividit en 4 capes, seguint la metodologia del patró "capes." La lògica d'aquest software és complexa és interessant salvaguardar-la i aïllar-la de la resta de l'aplicació, principalment de la presentació ja que tendeix a canviar. En el gràfic següent és mostra el funcionament de l'esmentada arquitectura.



L'arquitectura en capes permet separar la implementació en diferents funcionalitats, cada capa s'encarrega d'unes funcionalitats diferents. A grans trets s'explica que fa cada capa:

- **Capa Presentació** : És la que s'encarrega de la interacció amb l'usuari, és a dir, és des de on l'usuari es comunica amb el sistema i viceversa.

- **Capa Aplicació**: S'encarrega de transformar les dades de la *presentació* en objectes del *domini*, i les transfereix cap a la capa Domini. També fa la funció inversa, rep els objectes del domini i els prepara per a que la capa Presentació, en pugui utilitzar la informació dels objectes.

- **Capa Domini**: És la capa que conté els objectes del domini i la que es responsabilitza de comunicar les peticions de la capa Aplicació cap a la capa Persistència. Al inrevés, retorna els resultats del capa Persistència cap a la capa Aplicació.

- **Capa Persistència:** Aquesta capa s'encarrega de realitzar les ordres de consulta, modificació i inserció de la capa superior, en el llenguatge que usa la Base de Dades, el cas d'aquest projecte, fa la conversió de *java* a *Sql*.

Patrons de disseny

- Singleton

És un patró molt simple que ajuda a l'aplicació a no sobre carregar-se innecessàriament amb objectes instanciats varies vegades. Es fa servir quan en temps d'execució només és necessita una sola instància d'un objecte determinat. Aquest patró s'ha fet servir per exemple en la creació de les finestres de la capa presentació, en la connexió a la base de dades, el patró factoria i en la creació dels controladors.

- Factoria

L'objectiu d'aquest patró és separar la part del codi de creació d'objectes de la resta d'aplicació. Agafa sentit quan es creen objectes a partir d'interfícies, on la lògica és més complexa. El patró factoria s'ha fet servir en la classe "FactoriaRegistres".

- Indirecció

Amb aquest patró es millora el baix acoblament entre dues classes assignant responsabilitats de la mediació entre ells a un tercer. Per exemple, entre la classe "Vehicle" del domini i la classe "VehicleBD" de la persistència existeix una interfície "IVehicle" que assigna les responsabilitats de la intervenció entre elles i igual que amb aquestes dues classes, el patró indirecció

s'aplica a totes els capes de domini que estan relacionades amb la persistència.

- Controlador

El patró controlador és un patró que serveix com a intermediari entre una determinada interfície i l'algorisme que la implementa, de tal manera que és la rep les dades de l'usuari i les envia a les diferents classes segons el mètode cridat. Aquest patró suggereix que la lògica de negocis ha d'estar separada de la capa de presentació, això serveix per a augmentar la reutilització de codi i alhora tenir un control superior. Es recomana dividir els esdeveniments del sistema en el màxim nombre de controladors per a poder augmentar la cohesió i disminuir l'acoblament. Concretament en el programa els controladors es situen a la capa aplicació.

Seqüències

La majoria de taules utilitzen un identificador numèric, però no tots els ha d'inserir l'usuari sinó que ho fa automàticament la base de dades. El cas d'aquest projecte és interessant que les seqüències comencin a 1, i incrementin de un en un. Aquest problema el resol el mySQL amb una funció predefinida anomenada "AUTO_INCREMENT". La funció en si per defecte comença per 1 i incrementa de forma unitària. Els casos on es fa servir l'ús d'aquestes seqüències és a la taula banc, contacte, material, operació, tipusMaterial i tipusOperacio.

5. Implementació

És important recalcar que per optimitzar el software, es treballa en memòria i en base de dades, és a dir, quan es vol

recuperar informació primer es va al magatzem de dades, i si no es troba la informació demanada llavors es fa la consulta a la base de dades.

En l'apartat d'objectius s'ha vist que una fita important és que la interfícies sigui fàcil de fer servir i per tant la interfície és molt guiada. El que s'ha fet per evitar errors prematurs per equivocació alhora de prémer tecles i aconseguir que la interfície sigui usable, ràpida, fluïda, i alhora robusta.

Per evitar errors a l'usuari se li restringeix molt l'entrada de dades, és a dir, el software intenta proporcionar la màxima informació possible perquè l'usuari ompli menys camps.

Per seguir el patró controlador, s'ha implementat un controlador, per a cada subsistema, és a dir, hi ha el Controlador de Clients, que inclou totes les operacions relacionades amb els clients, Controlador Gestió Factures de Proveïdors que té implementades totes els mètodes necessaris per que funcioni aquest subsistema de l'aplicació. Els cada cop que és crida a un controlador es crida a una instància ja que utilitza el patró Singleton.

En l'apartat de disseny s'ha comentat que un objecte del domini no pot accedir a la base de dades directament sinó que ho fa mitjançant una interfície. Com que les interfícies no tenen constructor, no es pot fer un "new" de la interfície, i la persistència queda fora de l'abast de l'aplicació per fer un nou objecte d'una classe de la persistència. Utilitzant el `Class.forName("xxx").newInstance();`,

el que s'aconsegueix fer un recuperar una instància de la classe "xxx".

En la implementació de la persistència s'han creat classes mirall, és a dir, hi ha una classe per cada taula de la base de dades. Aquí s'implementen totes mètodes de per accedir a la base de dades. Per seguretat en les insercions es comprova que tot el procés ha funcionat correctament ja que sinó no fa el "commit" i la execució fa un "rollback".

6. Conclusions

L'aplicació que es presenta en aquest projecte final de carrera està pensada, dissenyada i elaborada per a un client en concret. Amb això es vol dir que altres empreses del sector segurament no la puguin utilitzar ja que s'ha desenvolupat en base a els processos concrets de Tallers Sacreu s.c.c.l.

D'altra banda es proposa una metodologia software pel desenvolupament de petits i mitjans sistemes d'informació. El temps ha estat el principal factor pel quan no s'ha pogut seguir fil per randa metodologies tan conegudes com Unified Proces o Mètrica v3.

7. Bibliografia

- I. Craig Larman, *UML y Patrones*, 2a Edición. Pearson, Prentice Hall.
 - II. Holzner, Steven, *la Biblia del JAVA 2*, Anaya Multimedia
 - III. <http://www.netbeans.org/kb/>, NetBeans Docs&Support, 2009
 - IV. <http://www.javahispano.org/>, Asociación JavaHispano, 2009
 - V. <http://www.lawebdelprogramador.com/>, La Web del Programador, 2009
-

Annex II- PSP (Procés Software Personal)

Paradigma escollit

Després de fer l'estudi de metodologies (revisar Annex) s'ha arribat a la conclusió que es farà servir el Procés Software Personal, que es un treball desenvolupat a l'assignatura de Desenvolupament de Sistemes d'Informació (DSI) i que fusiona idees de diversos models com Mètrica v3, UP de ideologia tradicional i de eXtreme Programing, una metodologia àgil.

Descripció PSP

És un model incremental i iteratiu, principalment basat en Unified Process. Me basat en UP (Unified Process) per diverses raons. La primera i la més important és que permet introduir nous requeriments al llarg del cicle de vida del projecte no cal que sigui al principi com en d'altres, per exemple els paradigmes de tipus seqüencial. Altres raons són que minimitza els riscos i obté arquitectures software més robustes perquè són provades com a mínim un cop per iteració, i que pràcticament des del primer moment tots el membres del projecte ja poden començar a treballar.

De Mètrica versió 3 també hi ha un parell d'idees interessants, el procés de planificació i el procés estudi de la viabilitat del sistema, però en el procés personal software s'han unit en un de sol. Per exemple en comptes de tenir planificació del SI i estudi de la viabilitat del sistema com a dos processos diferents, posar-los com "Avantprojecte" que només es fa al inici del projecte, abans de la primera iteració. Amb el que s'intenta aconseguir amb aquest "avantprojecte" es poder recollir o capturar els requeriments que fan servir en la primera iteració per poder començar a treballar, i poder fer els pressupostos i a calcular el temps que es dedica a cada iteració, o a cada cas d'ús.

Del model de procés software eXtreme Programming és interessant el feedback continu amb el client, estar fent "testing" sempre i la integració continua de codi. La darrera modificació del Unified Process és la ultima fase, la "Transition", tal i com es coneix només s'introdueix al final, ja que de cara a cada iteració es creu

innecessària. Per exemple, en les primeres iteracions no és necessari preparar els sistemes on s'han de fer les instal·lacions, migrar les dades dels sistemes antics, perquè això es pot fer al final de tot quan el projecte ja s'ha alliberat al client. Però per altra banda és veig útil a cada iteració, pel fet de realitzar les últimes proves i detecció d'errors, per examinar que els requeriments es compleixen satisfactòriament i per la preparació de documentació i manuals d'usuari.

ETAPES O PHASES

- Inception

En la primera etapa del procés personal software, el que es fa és reunir-se amb el client per tornar a buscar nous requeriments. En la primera iteració aquesta fase no falta que es faci perquè els requeriments inicials són els que s'han obtingut de l'avantprojecte. Però si que es pot anar fent el disseny i la implementació dels requeriments que ja es tenen.

- Elaboration

És una fase molt important perquè defineix l'arquitectura del sistema. En la fase d'elaboració es realitzen els diagrames de classes, els diagrames d'activitat, els diagrames de seqüència, els diagrames de casos d'ús, el model conceptual de la base de dades, en definitiva tots els models necessaris per saber que ha de fer i com es comportarà el sistema. En aquesta etapa torna a aparèixer el feedback amb el client, per tal d'ensenyar-li una mena de prototip dels requeriments captats fins al moment, amb això s'intenta obtenir-ne de nous o modificar algun requeriment mal captat. També és defineixen i es planifiquen les proves i la fase següent, la "Construction".

- Construction

L'objectiu d'aquesta fase és obtenir una versió beta del sistema amb totes les seves prestacions, llesta per poder-la provar. Per tant, es centra en la implementació de

tots el requeriments que hi ha a la fase d'elaboració. Integra el sistema i el prova tot alhora de forma exhaustiva, vigilant els detalls.

- **Transition**

Aquesta transició no és com la del model de procés software Unified Process, per això, és una Transition especial del procés software personal. En aquesta fase prepara la documentació del fet fins aleshores, corregir errors detectats en les proves realitzades en les fases anteriors, realitzar les darreres proves, examinar que els requeriments s'acompleixin satisfactòriament i planificar la següent iteració.

WORKFLOWS

Aquests són els workflows que s'inclouen en el procés personal software, la majoria es fan servir a cada iteració, però d'altres només es fan un sol cop, o al principi o al final.

Així doncs comencem pel principi, l'Avantprojecte.

- **Avantprojecte**

Aquest Workflow és el que es fa al inici del projecte i no es repeteix a cap iteració més.

- **Estudi de la viabilitat.**

És el primer que es fa quan s'inicia un projecte, estudiar si es possible o no portar a terme un projecte amb aquelles característiques. Un cop ja es sap que el projecte es pot dur a terme, és viable, es descriu la situació actual, es determina l'abast del sistema, ves valora el impacte a l'organització, la inversió necessària i així podem afinar més el preu alhora de fer el pressupost.

- **Captura dels primers requeriments.**

Arribat aquest punt hi ha un primer contacte amb els requeriments, funcionals i no funcionals. El client a de dir el que vol que faci el software. Un cop s'ha parlat amb el client es pot escriure un informe diferenciant els requeriments funcionals dels no funcionals.

- **Càlcul de temps de durada del projecte i el pressupost.**

Per calcular el temps i el pressupost orientatiu, es farà servir una mètrica de software estudiada a la assignatura de Gestió de Projectes Informàtics. Aquesta mètrica s'aplica calculant els Punts de Funció a partir del requeriments inicials, també es pot utilitzar CoCoMo un cop calculats els Punts de Funció, que donen la durada i l'esforç en les diferents fases del projecte, a partir d'aquí ja es pot calcular un pressupost orientatiu del projecte.

Principals artefactes

- Documentació de l'estudi de viabilitat amb la solució més adequada i l'informe de la situació actual i del què es vol millorar.
- Pressupostos

Requeriments

Aquest Workflow es repeteix tants cops com iteracions hi hagi en el projecte, excepte en la primera iteració perquè els requeriments ja els s'han recollit en l'Avantprojecte.

- **Reunió amb el client o feedback.**

Hi ha una reunió amb el client i se li mostra una versió beta amb els requeriments que es tenen fins aleshores, i que ho provi. D'aquesta manera pot dir si és el que realment espera o no, del programari que s'està desenvolupant.

- **Captura de més requeriments.**

A partir de la reunió amb client es poden treure conclusions, com si es va ben encaminat amb els requeriments o s'han de canviar o afegir d'altres. Per aquest motiu el feedback amb el client cobra especial importància.

Principals artefactes

- Informe de nous requeriments i/o de requeriments modificats.

- **Anàlisi & Disseny**

Aquest Workflow també es repeteix tants cops com iteracions tingui el projecte. En l'apartat de anàlisi i disseny hi afegit el model de casos d'ús i el model de domini, que en Unified Process es troba en el Workflow anterior, Requeriments.

Per dur a terme els diferents diagrames i models s'usaran d'eines RAD, com el Borland Together o el NetBeans que l'utilitzarem per fer el model de domini o diagrama de classes, el model de casos d'ús i els diagrames de seqüència tot això escrit en UML. Per fer els models de bases de dades es fa servir el PowerDesigner, de Sybase.

- **Anàlisi i Disseny del casos d'ús**

En aquest sub-apartat d'anàlisi i disseny s'escriuen els casos d'ús corresponents als requeriments funcionals que es tenen fins aleshores.

- **Anàlisi i Disseny de les classes**

En aquest altre sub-apartat s'ha de fer el diagrama de classes o el model de domini, cada classe ha de tenir els seus atributs i les seves corresponents relacions amb les altres classes.

- **Base de dades**

Aquí es fa el model conceptual de la base de dades, amb el PowerDesigner, que automàticament genera el model físic de la base de dades i l'Script de creació de la base

de dades, que posteriorment es fa servir. També s’han de deixar escrites les regles de negoci per a la fase de implementació poder-les codificar.

- Interfície gràfica d’usuari

S’ha de dissenyar la interfície gràfica d’usuari (GUI), és a dir, les pantalles amb que l’usuari interactuarà amb el sistema.

- Test Case

S’han de dissenyar els “tests case” de cada cas d’ús, indicant que s’està provant, les diferents entrades de dades i la resposta esperada del sistema per a cada dada introduïda.

Principals Artefactes

- Informe del casos d’ús.
- Model de casos d’ús.
- Diagrama de classes o model de domini.
- Diagrama de seqüència.
- Model conceptual de la base de dades.
- Model físic de la base de dades.
- Script de creació de la base de dades.
- Les diferents interfícies gràfiques d’usuari.
- Els “Tests Case”.

Implementation

Aquest Workflow es fa tants cops com iteracions tingui projecte. Principalment es porta a terme la implementació o codificació dels requeriments, utilitzant altre cop el Borland Together o altres eines, tota aquesta codificació s’escriurà amb el llenguatge Java, de Sun. També s’ha de crear la Base de Dades, amb l’editor Toad o mySQL o altres eines i sempre utilitzant SQL, d’Oracle.

- Implementar

En aquest apartat es codifica tant els requeriments com tot el que fa referència a la base de dades. En la codificació de requeriments s'utilitza una programació orientada a objectes, s'implementen les classes de cada "package" i els mètodes de cada classe. Referent a les Bases de Dades s'utilitza l'Script de creació obtingut del PowerDesigner, i sinó s'ha utilitzat el PowerDesigner, doncs s'ha d'escriure l'Script de creació. Indiferentment de si s'ha utilitzat l'eina de Sybase o no, també s'ha de codificar l'Script de les regles de negoci en l'editor de Bases de Dades.

- Revisar el codi implementat

Aquesta idea de Extrem Programming, revisa el codi, sense usar el compilador, per tal de buscar errors. En el cas de que existeixi algun error de codificació es torna a l'apartat anterior i es resol l'error.

- Compilar el codi

Compilació del codi per buscar error en temps d'execució, si es troba algun error es retorna a la implementació.

Principals artefactes

- Codi font.
- Script de regles de negoci.
- Script de creació de la base de dades (En cas de que no s'hagi obtingut abans.)

- **Test**

Aquest Workflow es porta a terme tants cops con iteracions té el projecte. Es torna a utilitzar el Borland Together o alguna altre eina per crear un programa executable de prova.

- Testing

Un cop està creat el programa de prova, recuperem els documents “Test case”, i es fan totes les proves pertinents. Si alguna prova falla es torna al Workflow anterior, i més concretament, a la implementació per corregir els errors detectats en el “testeig” del software.

Principals Artefactes

- Codi font i/o Scripts de les bases de dades corregits.
- Informe d’errors.

Transició

Possiblement no es el millor nom per aquest Workflow, ja que en el procés software personal també he considerat com una fase.

Aquest Workflow només es porta a terme en l’ultima iteració.

- Versió definitiva

Aquí s’ha d’obtenir una versió executable del software i preparar-la per poder instal·lar-la al maquinari del client.

- Instal·lar el programa al maquinari del client.

S’ha de preparar el maquinari perquè pugui suportar el software que posteriorment li serà instal·lat.

- Migració de dades

Es fa si i només sí, el client ja tenia un sistema d’informació anterior

Principals artefactes

- Versió executable
-

- **Manteniment del software**

Aquest Workflow només es fa un cop s'hagi lliurat el software al client.

Se li deixen un dies de prova, per si es detecta algun error. En aquest cas, en que el client detecti alguna errada té un període de dies preestablert per informar i el manteniment serà de franc, és un període de garantia.

Un cop passat aquest període de garantia el manteniment ja es cobra.

Cal matisar que s'entén per manteniment totes aquelles modificacions que no afectin a l'arquitectura del software.

Principals artefactes

- Actualitzacions de la versió definitiva, per tal de solucionar els canvis necessaris.

Mètrica Versió 3	Procés Software Personal
Planificació del SI	AvantProjecte
Estudi de la viabilitat del sistema	Requeriments
Anàlisi del SI	Anàlisi & Disseny
Disseny del SI	Implementació
Construcció del SI	
Implantació i acceptació del Sistema	Test
Manteniment del SI	Transició
	Manteniment del SI

Taula 6. Cuadre-Resum d'equivalències entre Mètrica v3 i el PSP

Unified Process	Procés Personal Software
Bussines Modeling	AvantProjecte
Requirements	Requeriments
Analysis	Anàlisi & Disseny
Design	Implementació
Implementation	
Test	Test
Deployment	Transició
	Manteniment del SI

Taula 7. Cuadre-Resum d'equivalències entre UP i el PSP

Annex III – Contingut del CD

- **Aplicació**
 - Codi font de l'aplicació (Netbeans IDE 6.5.1).

 - **Documentació**
 - Memòria del Projecte (Microsoft Office i Adobe Reader).
 - Portada de la memòria.
 - Diagrames UML i models de bases de dades.
 - Article del projecte (Adobe Reader).

 - **Fitxers d'instal·lació**
 - Fitxer d'instal·lació del Netbeans IDE 6.5.1.
 - Fitxer d'instal·lació de Sybase PowerDesigner 15.
 - Fitxer instal·lació mySQL.
 - Fitxer instal·lació SQLyog.
-

Bibliografia

- I. Craig Larman, *UML y Patrones*, 2a Edición. Pearson, Prentice Hall.
 - II. Holzner, Steven, *la Biblia del JAVA 2*, Anaya Multimedia
 - III. <http://www.netbeans.org/kb/>, NetBeans Docs&Support, 2009
 - IV. <http://www.javahispano.org/>, Asociación JavaHispano, 2009
 - V. <http://www.lawebdelprogramador.com/>, La Web del Programador, 2009
 - VI. <http://www.toedter.com/en/jcalendar/index.html>, Kai toedter, 2008
-