context confirmar(con:Integer)
post: 2.1 Si con = OK, es du a terme la còpia.
post: 2.2 sino, es finalitza el procés.

context ferEsborratXXX ()
post: 2.1 Es fa una petició d’esborrat XXX al Sistema.

context confirmar(con:Integer)
post: 2.1 Si con = OK, es du a terme l’esborrat.
post: 2.2 sino, es finalitza el procés.

6.5 Assignació de responsabilitats a capes

Un cop tenim els contractes i diagrames de seqüència normalitzats i amb la idea de com es durà a terme en la implementació, cal saber quines seran les responsabilitats que prendrà cada capa a l’hora de vigilar que es compleixin les pre condicions o post condicions que es requereixin.

Com comento a l’apartat de Descripció de l’Arquitectura Bàsica, la gestió de la persistència de les dades es fa mitjançant la estratègia de disseny directe, per la qual cosa la capa amb major nombre de responsabilitats passa a ser la capa de gestió de dades, mentre que les responsabilitats de la capa de domini queden reduïdes a, com a molt, dur a terme alguns càlculs necessaris entre la capa de gestió de dades i la capa de presentació.

En aquest cas, però, la capa de domini només servirà per fer senzilles manipulacions entre la capa de presentació i la capa de gestió de dades, el que significa aquelles operacions que podrien necessitar de la intervenció de la capa de domini, seran usades directament afegint atributs o alguna senzilla operació dins de la capa de presentació, o fent que hagi d’anar fins a la capa de gestió de dades.
Cal tenir en compte que el fet de tenir un sistema amb la capa de presentació separada, en part, de les altres dues capes, que tením en un altra part del sistema (utilitzem l’anomenada \textit{Presentació Distribuïda}), fa que l’accés a segons quines dades sigui més lent si la capa ha de accedir a nivells inferiors cada vegada que necessita uns paràmetres per a una operació molt senzilla.
També per aquest motiu, com la capa de domini està situada al mateix node que la capa de gestió de dades, el temps que triga es irrellevant.
A les operacions es veurà la feina que executa cada capa.

\begin{center}
\begin{tabular}{|c|}
\hline
Capa de presentació \tab \includegraphics[width=1cm]{capa1} \\
\hline
Capa de Domini \tab \includegraphics[width=1cm]{capa2} \\
\hline
Capa de Gestió de Dades \tab \includegraphics[width=1cm]{capa3} \\
\hline
SGDB/F \tab \includegraphics[width=1cm]{capa4} \\
\hline
PC's \tab \includegraphics[width=1cm]{pc} \\
\hline
\end{tabular}
\end{center}

\textbf{6.6 Operacions del disseny}

\textbf{6.6.1 Introducció a les operacions}

Les operacions d’aquesta sistema software en particular, seran implementades mitjançant el patró transacció, que permet executar una sèrie d’operacions des d’un moment donat a partir d’un esdeveniment en concret, com pot prèmer un botó o escollir una opció determinada en un desplegable.
També s’escull aquest patró perquè, donat que la capa de gestió de dades es la que controlarà la persistència de les dades, aprofitarem la potència que el SGBD en ofereixi, per això usarem el patró de disseny \textit{Passarel-la Fila}, que ens permet usar les pròpies sentències de la BD i utilitzar-les fàcilment, fent que el treball amb el SGBD sigui molt més intuitiu.
La idea de la feina de les capes es aquesta:
Llavors, la idea principal un cop estem a la capa de gestió de dades, es que puguem treballar amb SQL, en aquest cas, d'una manera senzill, i això és el que permet el patró abans esmentat:

Com es veu, aquest patró té els mateixos atributs que una taula de la BD, y cada instància correspon a una fila de la taula. Té operacions per afegir, modificar o esborrar files de la taula (contenen sentències SQL), i té el cercador, que es com a espècie de diccionari que permet trobar passarel·les partint de qualsevol atribut.

El patró transacció ens ofereix, també, una forma d'accedir a la BD sense intervenció de la capa de domini (quelcom que ens interessa) mitjançant la operació executar, que no té cap responsabilitat assignada però permet a d’altres crides a operacions accedir a la base de dades.
Cada classe representa la operació que es durà a terme, amb els paràmetres corresponents, i les operacions representen el retorn, amb booleans per notificar els errors i d’altres que ens interessin per a una operació determinada. La operació executar concreta, present a totes les classes, té la funcionalitat abans explicar de permetre a d’altres crides a funcions endinsar-se en el SGBD.

6.6.2 Operacions del sistema
6.6.2.1 Operació obteDades

Retorna nom, direcció, dni, cp i compte corrent del client de contracte amb codi
6.6.2.2 Operació ComprovarDades

6.6.2.3 Operació ÚltimesFactures

 Retorna nom, direcció, DNI i CP del client, i, de cada producte de la factura, el nom, la quantitat i el preu
6.6.2.4 Operació obteNom

Retorna el nom de tots els productes amb tipusMaterial = tipusMaterial

6.6.2.5 Operació Petició Material
6.6.2.6 Operació PeticioMaterial: extensió té-descricció

6.6.2.7 Operació PeticioIncidència

6.6.2.8 Operació bústiaSuggeriments
6.6.2.9 Operació liListatPeticioDades

Retorna el nClient, la data, la hora i la descripció de totes les instàncies de PetDades sense finalizar

6.6.2.10 Operació modificaClient

La capa de presentació s’ocuparà de que només s’actualitzin les columnes necessàries.
6.6.2.11 Operació AltaProducte

6.6.2.12 Operació novaExistencia

6.6.2.13 Operació actualitzaPreuProducte
6.6.2.14 Operació llistaProductesEstoc

Retorna codi, nom, preu, descripció i quantitat per tot producte amb quantitat \( \geq 1 \)

6.6.2.15 Operació llistaPetMat (\( * \) = igual per a les 3 opcions de processar una petició de material)
6.6.2.16 Operació executarPeticióM
6.6.2.17 Operació llistaPetMat (*): extensió producte-nou

6.6.2.18 Operació executarPeticióM: extensió producte-nou
6.6.2.19 Operació llistaPetMat (*): extensió producte-en-estoc

Frontera del sistema

6.6.2.20 Operació executarPeticioM: extensió producte-en-estoc

Frontera del sistema
6.6.2.21 Operació petMatClientN (**)

6.6.2.22 Operació executarPetMatClientN(***)

6.6.2.23 Operació petMatClientN (**): extensió específica-producte
6.6.2.24 Operació obtNom

Retorna el nom de tots els productes amb tipusMaterial = tipusMaterial

6.6.2.25 Operació específicaProducte

6.6.2.26 Operació executarPetMatClientN(***): extensió específica-producte
6.6.2.27 Operació PetMatClientN(**): extensió crea-producte

6.6.2.28 Operació creaProducteCodi
6.6.2.29 Operació llistaPrEsMin

Retorna codi, nom i descripció del producte

6.6.2.30 Operació altaComProp

Retorna codi, nClient, Hora i Data

6.6.2.31 Operació llProdComanda
6.6.2.32 Operació altaComanda

6.6.2.33 Operació lIPNoReb

Retorna nComanda, Quantitat i codi de producte de ComandaProveidor
6.6.2.34 Operació comandaPropRebuda

6.6.2.35 Operació llPNoServ

Retorna númeroComandaProveidor, ClientPr i productoDem

6.6.2.36 Operació producteServit
6.6.2.37 Operació IIRebudes

Retorna nComanda, Quantitat i codiP

6.6.2.38 Operació IIServeïdes

Retorna númeroComandaProveedor, ClientPr i producteDem

6.6.2.39 Operació histoPreus

Retorna llista de preu, data1 i data2
6.6.2.40 Operació altaServei

6.6.2.41 Operació modificarServei

Només modifica aquells paràmetres que li permet la capa de presentació

6.6.2.42 Operació llServeis

Retorna per tota instància de servei el nom, descripció i preu
6.6.2.43 Operació llistaPetIDNoLL

6.6.2.44 Operació executarPetID
6.6.2.45 Operació finalitza petID

6.6.2.46 Operació altaPagament

6.6.2.47 Operació afegirProd

Retorna la llista de productes i preu total parcial fins aquell producte
6.6.2.48 Operació altaPagament: extensió introduex-descripció

6.6.2.49 Operació llistaFCaixa(4*)

Retorna mes, any i setmana de totes les instàncies de FullCaixa

6.6.2.50 Operació llistaPagFCFac

Retorna el número de pagament, la forma, el preu i el número de pagament de tots el pagaments amb factura d’aquell full de caixa
6.6.2.51 Operació modPag

6.6.2.52 Operació llistaFCaixa(4*): extensió té-descrició
6.6.2.53 Operació llistaPagFCProd: extensió té-descrició

Retorna la descripció de productes, la forma, el preu i el número de pagament de tots els pagaments amb factura d’aquell full de caixa

6.6.2.54 Operació modPag:extensió té-descrició
6.6.2.55 Operació llistaPagTotals

Retorna sumEi (total de pagaments en efectiu), sumTa (total de pagaments amb tarja) i sumTot (suma de pagaments totals)

6.6.2.56 Operació nouAvis

Retorna, per cada avis no vist, el nAvis, aQui i la descripció
6.6.2.58 Operació processarAvisos

6.6.2.59 Operació llistaAvisosLlegits

Retorna, per cada avis vist, el nAvis, aQui i la descripció.
6.6.2.60 Operació nouInfTecnic

Retorna el nom, direcció, DNI, CP i província del client. A més a més, retorna els components, el pressupost i la data de l’informe tècnic.

6.6.2.61 Operació mostraInfTecnic

Retorna components, pressupost i data de l’informe tècnic.
6.6.2.62 Operació llistaIT

Frontera del sistema

executeIT() \[ \xrightarrow{\text{cercar}} \] \[ \text{lit} \]

obtenIT()

\[ \xrightarrow{\text{obtePressupost}} \]

\[ \xrightarrow{\text{obteData}} \]

\[ \text{dia} \]

\[ \text{nit} \]

\[ \text{ps} \]

\[ \text{dia} \]

\[ \text{x} \]

\[ \text{x} \]

Retorna lit, pressupost i data de tots els informes tècnics existents.

6.6.2.63 Operació suggCl

Frontera del sistema

executeIT() \[ \xrightarrow{\text{cercar}} \] \[ \text{Llegit = NO} \]

\[ \text{it} \]

\[ \text{psg: Pasarel la Sugeriments} \]

obteSuggeriment()

\[ \xrightarrow{\text{sugg}} \]

\[ \text{obteHora}() \]

\[ \text{hora} \]

\[ \text{obteData}() \]

\[ \text{data} \]

\[ \text{obtenClient}() \]

\[ \text{client} \]

\[ \text{x} \]

\[ \text{x} \]

Retorna el suggeriment, la hora, la data i el codi de client de tota instància no llegida de suggeriment.
6.6.2.64 Operació suggClVist

6.6.2.65 Operacions de copia de seguretat i esborrat de dades obsoletes del sistema.

Aquests apartats tenen la diferència que tracten amb la base de dades de manera diferent, no treballen amb sentències SQL, si no que copien o esborren les dades de manera directa al disc dur de seguretat, en el cas de les còpies, o esborren en el disc dur principal, si no hi ha còpies de seguretat, o en ambdós si existeixen aquestes còpies.
Copia de seguretat

Copia de seguretat

Disc Dur Principal

GUARDAR

Disc Dur Seguretat

:TxEsborra

executar() -> cerca(tots_atributs XXX)

esborra_files_HDD(cp_t)

esborra_files_HDD_sec(cp_t)

Frontera del sistema

:CercadorTaula

132
6.7 Taules de la Base de Dades

6.7.1 Introducció
En aquest apartat descriuré una mica la nomenclatura utilitzada per descriure la base de dades que utilitzo i les corresponents taules.
En negreta tenim el nom de la taula, que té força correspondència amb les classes amb el diagrama estàtic del model conceptual de les dades, tot i que hi ha petites diferències, relacionades amb la implementació, sobretot.
Dins els parèntesis tenim els atributs dels quals es compon cada objecte, on podem tenir, també, 3 tipus diferents:
✓ **Clau primària**, que es l’atribut que diferència un objecte concret d’una taula de tots els altres.
✓ **Clau forana**, que es un atribut que relaciona aquella objecte amb la clau primària d’un altre objecte. Per exemple, que una petició sigui per un objecte determinat, i puguem saber quin objecte es.
✓ **Clau primària i forana a la vegada**, que obligatòriament ha de ser una taula que tingui més d’una clau primària, perquè si no donaria problemes a l’utilitzar les dues taules simultàniament.

6.7.2 Taules de la base de dades
**Avisos** (nAvis, aQui, Descripcio, Vist, Data, Hora)
**Client** (nClient, Nom, DNI, Direccio, CP, Telefon, Poblacio, Provincia, eMail, Observacions, CC, te_contracte)
**ComandaPropia** (nComanda, codiPr, nProve, Rebuda)
**ComandaProveidor** (numeroProducteComanda, Servida, clientPr, producteDem, nProv)
  ✓ clientPr fa referència a a Client
  ✓ producteDem fa referència a a Producte
  ✓ nProv fa referència a a Proveïdor

**Factura** (numFactura, pagada, preuTotal, formaPagament, dataCreacio, dataVenciment, nClient)
  ✓ dataCreacio fa referència a a Data (taula no representada)
  ✓ dataVenciment fa referència a a Data (taula no representada)

**FullCaixa** (Setmana, Mes, Any)

**Històrial Preus** (Codi, Data, dataFinal, preuCompra)
  ✓ Codi fa referència a a Producte
  ✓ Data fa referència a a Data (taula no representada)

**Incidència** (numeroIncidencia, Descripció, Observacions, estatIncidencia, Desplaçament, esRep, dataActuacio, nTec, nClient)
  ✓ esRep fa referència a a Data (taula no representada)
  ✓ dataActuacio es fa referència a a Data (taula no representada)
  ✓ nTec fa referència a a Tècnic
  ✓ nCliente fa referència a a Client

**Informe Tècnic** (nIT, Components, Pressupost, Client, Data)
  ✓ Client fa referència a a Client
  ✓ Data fa referència a a Data (taula no representada)

**OR** (numeroOR, Descripcio, Formateig, Problemes, estatOR, entradaOR, nClient)
  ✓ entradaOR fa referència a a Data (taula no representada)
  ✓ nClient fa referència a a Client

**PagFactura** (nPagament, Setmana, Mes, Any, formaPagament, preuPag, Numero)
  ✓ Setmana, Mes i Any fa referència a a FullCaixa

**PagProducte** (nPagament, Setmana, Mes, Any, formaPagament, preuPag, Descripcio)
  ✓ Setmana, Mes i Any fa referència a a FullCaixa

**PetDades** (nClient, Data, Hora, Descripcio, Finalitzada)
  ✓ nClient fa referència a a Client
  ✓ Data fa referència a a Data (taula no representada)
  ✓ Hora fa referència a a Hora (taula no representada)

**PetID** (nClient, Data, Hora, Descripcio, Processada)
  ✓ nClient fa referència a a Client
  ✓ Data fa referència a a Data (taula no representada)
  ✓ Hora fa referència a a Hora (taula no representada)

**PetMaterial** (nClient, Data, Hora, Demanat)
  ✓ nClient fa referència a a Client
  ✓ Data fa referència a a Data (taula no representada)
  ✓ Hora fa referència a a Hora (taula no representada)

**PmDescrit** (nClient, Data, Hora, Demanat, Descripcio)
  ✓ nClient fa referència a a Client
  ✓ Data fa referència a a Data (taula no representada)
  ✓ Hora fa referència a a Hora (taula no representada)
Producte (Codi, Descripcio, preuVenta, estocMinim, Quantitat, tipusProducte, nPro)
✓ nPro fa referència a a Proveïdor
ProducteComanda(nClient,Data, Hora, Codi, donatdAlta)
✓ nClient fa referència a a Client
✓ Data fa referència a a Data (taula no representada)
✓ Hora fa referència a a Hora (taula no representada)
✓ Codi fa referència a a Producte
ProducteFactura(codi, numFactura, Quantitat, PreuFactura)
✓ codi fa referència a a Producte
✓ numFactura fa referència a a Factura
Proveïdor (nProveïdor, Nom, DNI)
Suggeriments (nClient, Data, Hora, suggerimentClient, Llegit)
✓ nClient fa referència a a Client
✓ Data fa referència a a Data (taula no representada)
✓ Hora fa referència a a Hora (taula no representada)
Tècnic (nTècnic, Nom, DNI)
tipusDeServei (Codi, Descripcio, Nom, preuServei)

6.7.1 Índex i millora de l’access a les dades de la Base de Dades
Per implementar la base de dades usarem MySQL dins de PHPMyAdmin, es a dir, un software que em permet gestionar una base de dades relacional en SQL, però enfocada a l’ús amb el llenguatge PHP, que es el que he usat per obtenir i gestionar les dades que obtinc mitjançant els formularis Web que utilitzo.
Aquesta BD optimitza per ella mateixa els accessos a les dades, permeten que les taules amb més accessos siguin més accessibles que aquelles que no s’utilitzen tant, a nivell estadístic (SELECTs, DELETEs,..)
A més a més, deixa de la mà de l’usuari la possibilitat d’optimitzar-la encara més eliminant els residus que hi pogués deixar.
Segons la informació que ens ofereix el manual de MySQL, en un ordinador Pentium II a 400Mhz, es triguen a fer un milió d’operacions d’addició simple en 0.32 segons. Feta la mateixa prova en el meu ordinador (~Pentium IV a 3500Mhz), el temps d’accés es redueix fins els 0.033 segons, e inclòs si l’accés afecta les taules, arriba a un màxim de 0.8 segons, que segueix sent molt inferior. Els ordinadors que disposa EBV tenen una capacitat més propera als 2000Mhz, per tant l’accés serà força alta de totes maners.
A més de la pròpia optimització de que disposa MySQL, tenim la possibilitat de millorar la indexació, amb mètodes de cerca basats en el mergedsort, el conegut algoritme de cerca que divideix la taula recursivament, i la va ordenant. En el nostre cas particular, no ens cal usar aquests algoritmes perquè el volum de dades a tractar en El Byte Veloç no necessita d’una optimització especialment alta, ja que tenim un accés prou bo, però va bé saber quines possibilitats de millora tenim en un futur.