

# **Disseny i implementació d'un sistema per a gestionar la postvenda d'electrodomèstics**

**Antoni Garriga Rovira**

Grau d'Enginyeria Informàtica

**Josep Vallverdú Vergé**

Divendres, 10 de juny de 2011



## Resum

Aquest treball se centra en el disseny i implementació d'una base de dades relacional per a la gestió d'incidències de productes, atenent als requeriments inicialment establerts.

S'inicia amb una introducció on es fa la justificació del treball, s'especifiquen els objectius, l'enfocament, el mètode seguit, la planificació establerta, els productes obtinguts i es descriuen breument la resta de capítols del treball. Segueix el capítol per la preparació del programari necessari per realitzar-lo, sobretot en la part més pràctica. Es detalla el programari bàsic utilitzat en el treball, i en els casos més importants, com configurar-lo o preparar-lo.

En la part pràctica, primer es realitza la base de dades operacional (BD) amb els procediments emmagatzemats per la gestió i el sistema de diari (log), que ens permetrà guardar les accions fetes a la BD i que podem utilitzar per resoldre problemes potencials que puguin sorgir. Tot seguit es crea el magatzem de dades (DWH), així com els processos d'extracció, transformació i càrrega (ETL) pel traspàs d'informació entre la BD operacional i el DWH. I finalment es realitzen els informes per extreure la informació del DWH, que es complementa amb la realització d'una interfície per navegador web que permet realitzar les consultes dels informes al DWH, i consultar les taules més rellevants de la BD i el DWH.

El treball finalitza amb una valoració econòmica del projecte i les conclusions que en podem extreure, on s'indica on hem pogut arribar després de fer el treball, què hem constatat, observat o trobat.

# Índex de continguts

<b>Resum.....</b>	<b>1</b>
<b>Índex de continguts.....</b>	<b>2</b>
<b>Índex de figures.....</b>	<b>5</b>
<b>1.Introducció.....</b>	<b>7</b>
1.1.Justificació del TFG i context en el qual es desenvolupa: punt de partida i aportació del TFG.....	7
1.2.Objectius del TFG.....	8
1.3.Enfocament i mètode seguit.....	9
1.4.Planificació del projecte.....	9
1.4.1.Tasques.....	10
1.4.2.Fites.....	10
1.4.3.Temporització de les tasques.....	11
1.5.Productes obtinguts.....	11
1.6.Descripció dels capítols que segueixen.....	12
<b>2.Preparació del programari.....</b>	<b>13</b>
2.1.RISE, diagrames ER.....	13
2.2.PostgreSQL, SGBD.....	14
2.3.Apache Tomcat, servidor web.....	17
2.4.Java, llenguatge de programació del servidor web.....	19
2.5.Prova del funcionament entre el servidor web i el SGBD.....	21
2.5.1.Creació d'una BD a PostgreSQL.....	21
2.5.2.Preparació servidor Tomcat.....	21
2.5.3.Execució i resultat.....	22
<b>3.Base de dades operacional.....</b>	<b>23</b>
3.1.Creació de la BD.....	23
3.1.1.Disseny de la BD.....	23
3.1.2.Scripts de creació de la BD.....	27
3.2.Sistema de diari (log).....	28
3.3.Procediments emmagatzemats.....	29
3.4.Prova de càrrega de dades a la BD.....	29

<b>4. Magatzem de dades</b>	<b>30</b>
4.1. Creació del magatzem de dades	30
4.1.1. Requisits	30
4.1.2. Anàlisi i disseny	30
- Model conceptual	31
- Model físic	32
4.1.3. Implementació	32
4.1.4. Proves	33
4.2. Processos ETL (extracció, transformació i càrrega)	33
4.2.1. Extracció de dades de la BD operacional	33
4.2.2. Transformació de les dades	35
4.2.3. Càrrega de dades al DWH	35
<b>5. Consultes al magatzem de dades</b>	<b>38</b>
5.1. Nombre d'incidències per tipus d'incidència	38
5.2. Nombre d'incidències per tipus de producte	38
5.3. Cost de les garanties per tipus d'incidència	38
5.4. Cost de les garanties per tipus de producte	39
5.5. Cost de les incidències en un determinat període	39
5.6. Cost de les garanties en un determinat període	39
<b>6. Consultes des d'un navegador web</b>	<b>40</b>
6.1. Pàgines web del client	40
6.1.1. Menú inicial	41
6.1.2. Menú informes DWH	42
6.2. Processos del servidor	42
6.2.1. Configuració del servidor web	43
6.3. Accés a la BD i al DWH a través del navegador	43
<b>7. Valoració econòmica del projecte</b>	<b>44</b>
7.1. Cost del maquinari	44
7.2. Cost del programari	44
7.3. Cost del desenvolupament	45
<b>8. Conclusions</b>	<b>46</b>
<b>Glossari</b>	<b>48</b>
<b>Bibliografia</b>	<b>50</b>

<b>Annexos.....</b>	<b>52</b>
Annex 1: Diagrama de Gantt de la temporització de tasques.....	52
Annex 2: Codi del programa Java 'MostraBD.java'.....	53
Annex 3: Codi SQL per la creació de les taules de la BD.....	55
Annex 4: Procediment emmagatzemat de la taula 'accio'.....	58
Annex 5: Codi SQL per la creació de les taules del DWH.....	60
Annex 6: Procediments emmagatzemats de la taula 'DWH_acc'.....	61
Annex 7: Pàgines HTML per accedir a la BD i al DWH .....	63
• MenuInici.html.....	63
• MenuInformes.html.....	64
• EntradaDatesIncid.html.....	65
Annex 8: Detall del codi del programa Java 'ElectroDWH.java'.....	66
Annex 9: Fitxer de configuració del servidor web 'web.xml'.....	69
Annex 10: Relació de fitxers adjunts a la memòria del treball.....	70

## Índex de figures

### Taules

Taula 1.1: Fites de la planificació.....	10
Taula 1.2: Temporització de les tasques planificades.....	11
Taula 2.1: Variables d'entorn necessàries pel programari Java.....	20
Taula 4.1: Atributs de les dimensions model conceptual DWH.....	31
Taula 7.1: Cost del maquinari.....	44
Taula 7.2: Cost del programari.....	45
Taula 7.3: Cost del desenvolupament.....	45
Taula A10.1: Relació de fitxers adjunts a la memòria del treball.....	70

### Il·lustracions

Il·lustració 2.1: Pàgina web per descarregar el programari RISE Editor.....	13
Il·lustració 2.2: Pàgina web per descarregar el programari PostgreSQL.....	14
Il·lustració 2.3: Procés instal·lació del programari PostgreSQL.....	15
Il·lustració 2.4: Pàgina web per descarregar documentació PostgreSQL.....	15
Il·lustració 2.5: Crear una base de dades amb "pgAdmin III".....	16
Il·lustració 2.6: Executar un script SQL amb "pgAdmin III".....	16
Il·lustració 2.7: Versions programari Apache Tomcat.....	17
Il·lustració 2.8: Distribucions Apache Tomcat segons sistema operatiu.....	17
Il·lustració 2.9: Estructura de directoris Tomcat.....	18
Il·lustració 2.10: Pantalla d'arrancada del Tomcat.....	18
Il·lustració 2.11: Pàgina web de connexió a Apache Tomcat.....	19
Il·lustració 2.12: Pàgina web per descàrrega el programari Java.....	19
Il·lustració 2.13: Pàgina web per descarregar el JDBC.....	20
Il·lustració 2.14: Codi SQL creació de la base de dades mydatabase.....	21
Il·lustració 2.15: Resultat consulta SQL de la base de dades mydatabase.....	21
Il·lustració 2.16: Codi web.xml per prova de funcionament del Tomcat.....	22
Il·lustració 2.17: Resultat prova de consulta al servidor web.....	22
Il·lustració 3.1: Disseny conceptual de la BD.....	23
Il·lustració 3.2: Atributs del disseny conceptual.....	25

Il·lustració 3.3: Relacions del model ER.....	26
Il·lustració 3.4: Diagrama ER de la BD.....	27
Il·lustració 3.5: Codi SQL de la creació de la BD "electrodomestics".....	27
Il·lustració 3.6: Codi SQL de la creació de la taula 'log'.....	28
Il·lustració 3.7: Funció 'gravar_log' per centralitzar gravacions al diari.....	28
Il·lustració 3.8: Registres gravats a la taula 'log'.....	28
Il·lustració 3.9: Consulta de productes a la BD.....	29
Il·lustració 4.1: Estrella bàsica.....	31
Il·lustració 4.2: Dimensió Temps.....	31
Il·lustració 4.3: Esquema relacional.....	32
Il·lustració 4.4: Diagrama ER del DWH.....	32
Il·lustració 4.5: Dades carregades al magatzem de dades.....	37
Il·lustració 5.1: Consulta 1 al DWH.....	38
Il·lustració 5.2: Consulta 2 al DWH.....	38
Il·lustració 5.3: Consulta 3 al DWH.....	38
Il·lustració 5.4: Consulta 4 al DWH.....	39
Il·lustració 5.5: Consulta 5 al DWH.....	39
Il·lustració 5.6: Consulta 6 al DWH.....	39
Il·lustració 6.1: Mapa web.....	40
Il·lustració 6.2: Estructura de directoris aplicació Tomcat.....	41
Il·lustració 6.3: Web, Menú inicial.....	41
Il·lustració 6.4: Web, menú informes DWH.....	42
Il·lustració 6.5: Web, pàgina per especificar període de consulta.....	42
Il·lustració 6.6: Consultes per interfície web.....	43
Il·lustració A1.1: Diagrama Gantt de la planificació del treball.....	52



# 1. Introducció

En aquest primer capítol de la memòria del treball es fa una introducció justificant el TFG, especificant els objectius, explicant l'enfocament, el mètode seguit, la planificació establerta, els productes obtinguts i, finalment, es descriuen breument la resta de capítols d'aquesta memòria.

## 1.1. Justificació del TFG i context en el qual es desenvolupa: punt de partida i aportació del TFG

El projecte realitzat en aquest treball fi de grau (TFG) s'emmarca en l'àrea de les bases de dades i s'han posat en pràctica els coneixements adquirits d'aquesta àrea, consolidant-los i ampliant-los emprant un sistema de gestió de bases de dades (SGBD) relacional amb el qual no s'hi ha treballat mai abans, i emprant el llenguatge PL/SQL i SQL dinàmic.

En concret, aquest projecte se centra en el disseny i implementació d'una base de dades relacional (BD) per a la gestió d'incidències en la postvenda de productes d'una empresa especialitzada en la fabricació d'electrodomèstics, atenent les especificacions inicialment establertes, que són:

- Implementar una BD relacional que satisfaci les necessitats de l'empresa, i on tota la gestió i manteniment de la informació es faci únicament mitjançant procediments de base de dades.
- La BD ha de permetre la gestió de les incidències en la postvenda dels productes de l'empresa, concretament per a cada producte ha de mantenir l'estat actual i els passos que s'han fet per a resoldre la incidència. Ha de tenir totes les dades necessàries per a la correcta gestió, com són: els productes, els agents telefònics, els enginyers de suport... Així mateix, ha de ser escalable per poder incorporar les noves necessitats que sorgeixin en el futur.
- A part de la BD operacional, cal definir un magatzem de dades, o *datawarehouse* (DWH), per consolidar les dades necessàries per poder generar estadístiques que ajudaran a millorar l'eficiència de l'empresa. Així mateix, cal implementar els processos de traspàs d'informació entre la BD i el DWH, i les consultes bàsiques necessàries.

- Per facilitar el manteniment del sistema, es generaran mecanismes per poder resoldre problemes d'integració, que es concreten en un diari (*log* en anglès) de les accions que es fan a la BD, i mecanismes per testejar les funcionalitats de la mateixa BD.

Aquest TFG ens aporta una solució detallada, a les moltes possibles que hi pot haver, a la situació inicialment plantejada.

## 1.2. Objectius del TFG

Com a objectius bàsics requerits del projecte es marquen els següents:

- Detectar les necessitats bàsiques del sistema a implementar.
- Detectar possibles funcionalitats addicionals de valor afegit.
- Proposar un disseny que s'ajusti als requeriments exposats.
- Implementar el sistema encapsulant les funcions d'accés a les dades.

Com a objectius específics d'aquest projecte es defineixen els següents:

- Treballar amb un SGBD relacional amb el qual no s'hi ha treballat mai abans.
- Fer l'anàlisi i el disseny de la BD, els scripts de creació necessaris per a crear-la i implementar els procediments emmagatzemats necessaris perquè el sistema funcioni i les aplicacions els utilitzin.
- Analitzar i definir un DWH per a emmagatzemar la informació històrica de la BD que interressi guardar per a realitzar-hi consultes eficientment.
- Implementar els processos de traspàs de la informació entre la BD i el DWH, i realitzar una sèrie de consultes predefinides per als usuaris.
- Preparar una interfície amb HTML amb el menú de funcionalitats que podran fer els usuaris per consultar a través d'un servidor web la BD i el DWH.
- Utilitzar el Java com a llenguatge de programació que s'executarà al servidor web per centralitzar l'accés al SGBD.
- Implementar un sistema de diari (*log*) detallat de les accions fetes a la BD per poder detectar problemes que puguin aparèixer.
- Preparar un joc de proves per provar i testejar el sistema.
- Documentar la implementació i el funcionament.

### 1.3. Enfocament i mètode seguit

A l'inici es va enfocar el treball al mètode *Scrum*, que és un mètode de desenvolupament àgil, iteratiu i incremental que permet canviar les idees inicials a mesura que el projecte avança, que facilita el descobriment de la solució mitjançant cicles de retroalimentació, i que es basa en la teoria empírica de control de processos, emprant un enfocament iteratiu i incremental per optimitzar la previsibilitat i el control de riscos.

Es va fer un estudi de la metodologia *Scrum*, i l'idea inicial era adaptar les parts del producte a realitzar amb les entregues de parcials del treball i amb les iteracions dels mètode (*Sprint backlog*). Veient que en les primeres fases del treball s'encaraven més cap al mètode en cascada, per la falta d'experiència en aquest mètode àgil, i que no era objecte d'aquest treball aprendre i aplicar una nova metodologia, finalment es va enfocar a metodologia en cascada, ja que les característiques del treball s'hi adaptaven millor, perquè teníem un objectiu clar i la solució era coneguda, no es preveia variar els requeriments durant l'evolució del projecte.

Així, en seguir el mètode en cascada, o cicle de vida clàssic, hem aplicat les diferents etapes del mètode, que són: la definició dels requisits, per definir quin ha de ser el producte a desenvolupar; l'anàlisi i disseny, per establir com ha de ser el producte tant des del punt de vista extern com intern; la implementació, per generar el producte; les proves, per verificar que el producte desenvolupat correspon als requisits definits; i el manteniment, per corregir el defectes, o aplicar les millores, que es vagin trobant.

### 1.4. Planificació del projecte

Per fer la planificació del projecte per desenvolupar el treball ens hem basat en dividir el projecte en diferents tasques. Les tasques s'han planificat de manera seqüencial, i s'han ajustat amb les fites definides, que ens han vingut indicades per les dates d'entrega parcial (PAC) que ja estaven preestablertes al calendari, i sobretot amb la data de lliurament final d'aquest treball. Cada tasca s'ha desenvolupat seguint la metodologia indicada, documentat les diferents accions fetes, i al final de cada una s'ha obtingut una part del producte.

### 1.4.1. Tasques

Les tasques que s'han definit i que posteriorment s'han realitzat, segons les funcionalitats o requisits previstos del producte, són les següents:

1. **Preparació del programari.** Recerca d'informació, obtenció i instal·lació dels programes necessaris per poder realitzar el treball segons els requeriments establerts.
2. **Base de dades operacional.** Anàlisi dels requeriments, disseny i implementació de la BD per satisfer les necessitats que s'han establert com a base d'aquest treball.
3. **Magatzem de dades.** Anàlisi dels requeriments, disseny i implementació del magatzem de dades per consolidar les dades necessàries amb els processos de traspàs d'informació entre la BD i el DWH, carregant la informació històrica necessària.
4. **Consultes al magatzem de dades.** Implementació de les consultes bàsiques necessàries al magatzem de dades per obtenir les dades estadístiques necessàries.
5. **Consultes des d'un navegador web.** Preparació d'una interfície centralitzada perquè els usuaris puguin realitzar de manera fàcil i transparent les consultes al magatzem de dades, i a dades rellevants de la BD, a través d'una pàgina web.
6. **Lliurament final.** Realització de les correccions finals, preparació del lliurament de la documentació d'aquest treball.

### 1.4.2. Fites

Les fites en què hem basat el treball han estat les marcades per les entregues parcials i el lliurament final, i dins el període de cada fita hem fet coincidir la finalització d'algunes tasques, com veiem en la següent taula:

Fita	Data	Tasques finalitzades a la fita
PAC2	17/04/2011	Preparació del programari. Base de dades operacional.
PAC3	15/05/2011	Magatzem de dades. Consultes al magatzem de dades.
Lliurament final	12/06/2011	Consultes des d'un navegador web. Lliurament final.

Taula 1.1: Fites de la planificació

### 1.4.3. Temporització de les tasques

La temporització de les tasques es va fer inicialment en el pla de treball, preveient-les de manera seqüencial, i en el decurs del treball s'ha ajustat amb petites variacions. El que no ha tret que en alguns casos s'ha aplicat el manteniment de tasques ja finalitzades per aplicar millores en algunes parts, tal i com ens permet el mètode en cascada en la fase de manteniment.

En la següent taula es detalla la temporització en un calendari on s'han considerat només els dies laborables de dilluns a divendres, tot i que com es va preveure en el pla de treball, els dissabtes o festius s'han emprat per completar les tasques que no s'havien pogut acabar en els dies previstos:

N.	Tasca	Data inici	Data final	Durada
1	Preparació del programari	21/03/2011	25/03/2011	5 dies
2	Base de dades operacional	28/03/2011	15/04/2011	15 dies
3	Magatzem de dades	18/04/2011	06/05/2011	15 dies
4	Consultes al magatzem de dades	09/05/2011	13/05/2011	5 dies
5	Consultes des d'un navegador web	16/05/2011	20/05/2011	5 dies
6	Lliurament final	23/05/2011	10/06/2011	15 dies

*Taula 1.2: Temporització de les tasques planificades*

En l'annex 1 s'hi adjunta el diagrama de *Gantt* de la temporització, que complementa i amplia la informació.

## 1.5. Productes obtinguts

Memòria del treball.

Aquest document on hi queda reflectida tota la feina feta en el treball.

Presentació.

Resum del treball.

Treball pràctic.

Inclou tot el producte desenvolupat, concretament:

Base de dades operacional.

Scripts de creació de les taules bàsiques de la BD.

Scripts de creació de la taula del diari.

Procediments de la BD.

Comandes per carregar dades per fer el test de proves.

Comandes per extreure la informació per al DWH.

Magatzem de dades.

Scripts de creació del DWH.

Procediments del DWH.

Comandes per carregar dades per fer el test de proves.

Comandes per carregar les dades extretes a la BD.

Consultes al magatzem de dades.

Comandes per obtenir la informació estadística del DWH.

Consultes des d'un navegador web.

Codi HTML del client i del servidor web.

Codi Java del servidor web que accedeix al SGBD.

## 1.6. Descripció dels capítols que segueixen

A partir d'aquest primer capítol introductori al treball, segueixen els següents capítols:

- **Preparació del programari.** On es detalla el programari utilitzat, com aconseguir-lo, i com configurar-lo o preparar-lo per deixar-lo a punt per al treball.
- **Base de dades operacional.** On construïm la base de dades segons els requeriments.
- **Magatzem de dades.** On construïm el magatzem de dades per carregar les dades històriques necessàries de la BD.
- **Consultes al magatzem de dades.** On creem les consultes per explotar el DWH.
- **Consultes des d'un navegador web.** On construïm una interfície amb HTML, client i servidor, amb les consultes que podran fer els usuaris per consultar a través d'un servidor web el DWH i la part relacionada de la BD.
- **Valoració econòmica del projecte.** On fem un estudi del cost del projecte.
- **Conclusions.** Amb les conclusions que extraïem del treball.

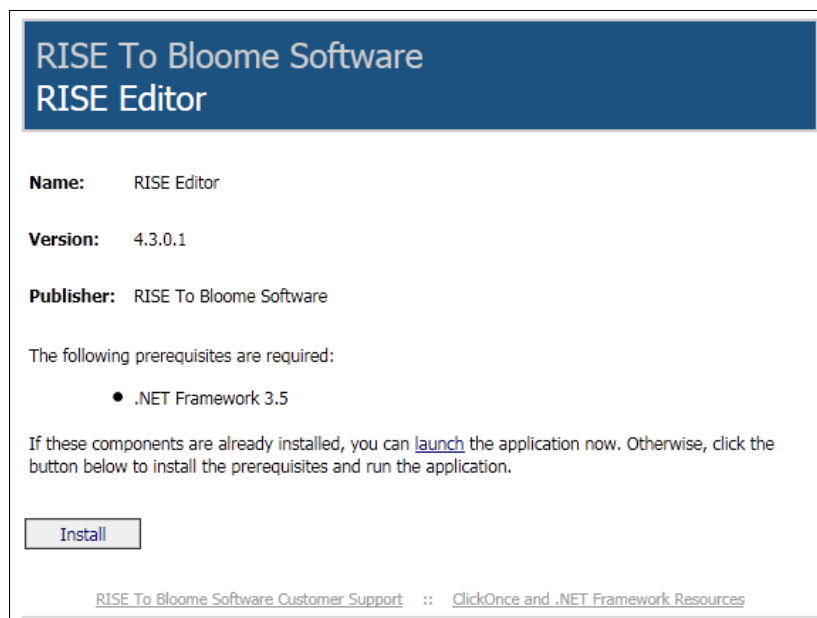
## 2. Preparació del programari

En aquest capítol inicial de la memòria del treball es detalla el programari bàsic utilitzat en el treball en sistema operatiu "MS Windows Vista", com aconseguir-lo, i en els casos més importants, com configurar-lo o preparar-lo per deixar-lo a punt per al treball.

### 2.1. RISE, diagrames ER

Per a realitzar els diagrames entitat-relació (ER) per representar les bases de dades s'ha utilitzat el programari *RISE Editor*<sup>1</sup>, que és una eina innovadora per al desenvolupament de models de solucions d'informació en una interfície visual i intuïtiva. Com a característiques del model ER ens permet: múltiples diagrames, l'edició totalment gràfica, notació UML, anotacions, elements compostos...

S'ha utilitzat la versió 4.3.0.1, la darrera versió disponible en fer el treball en el web <http://www.risetobloome.com/clickonce/riseeditor/riseeditor.htm>.



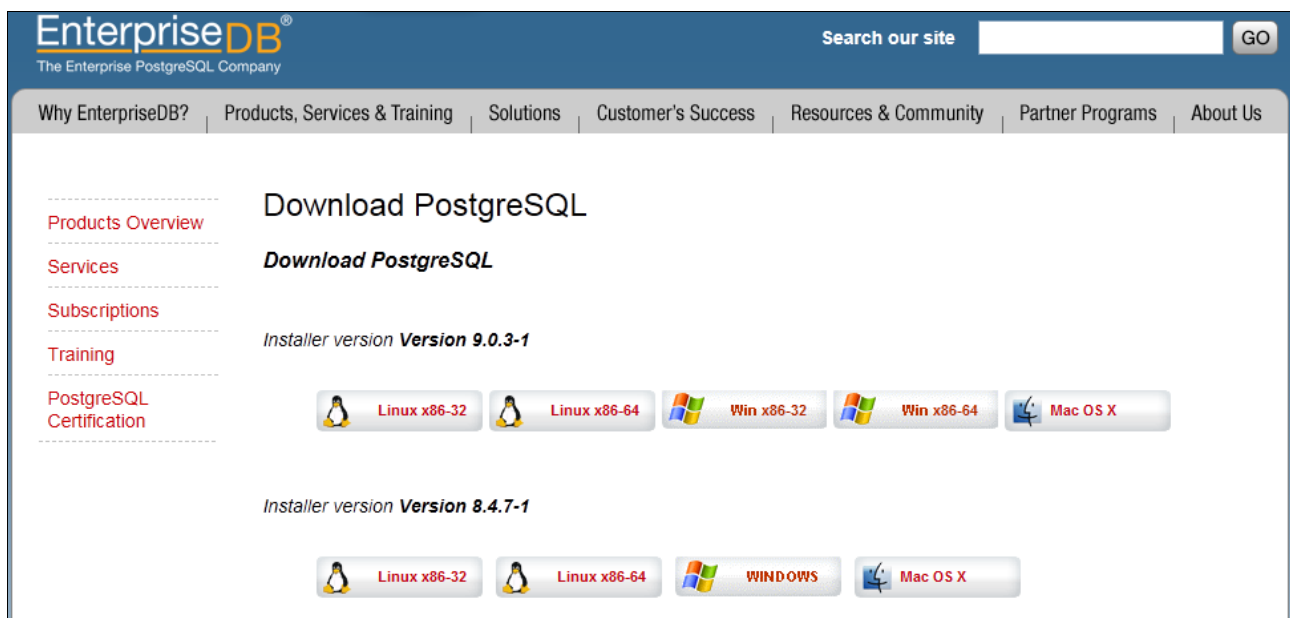
Il·lustració 2.1: Pàgina web per descarregar el programari RISE Editor

1 De l'empresa RISE to Bloom, empresa innovadora de programari amb seu a Uppsala, Suècia.

## 2.2. PostgreSQL, SGBD

*PostgreSQL* és un potent sistema de base de dades objecte-relacional de codi obert. Es pot executar en molts dels principals sistemes operatius, com Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) i Windows. Compleix totalment les propietats ACID (*atomicity, consistency, isolation, durability*) que garanteixen que les transaccions de base de dades es processen de forma fiable, té suport complet per claus foranes, unions, vistes, disparadors i procediments emmagatzemats (en diversos llenguatges). Inclou la majoria de tipus de dades segons SQL: 2008, que és la sisena revisió de la norma ISO i ANSI per al llenguatge de consulta SQL de base de dades, com són: *INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, i TIMESTAMP*. També suporta emmagatzemament d'objectes binaris grans, com imatges, sons o vídeo. Disposa de interfícies natives de programació per a C / C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre altres.

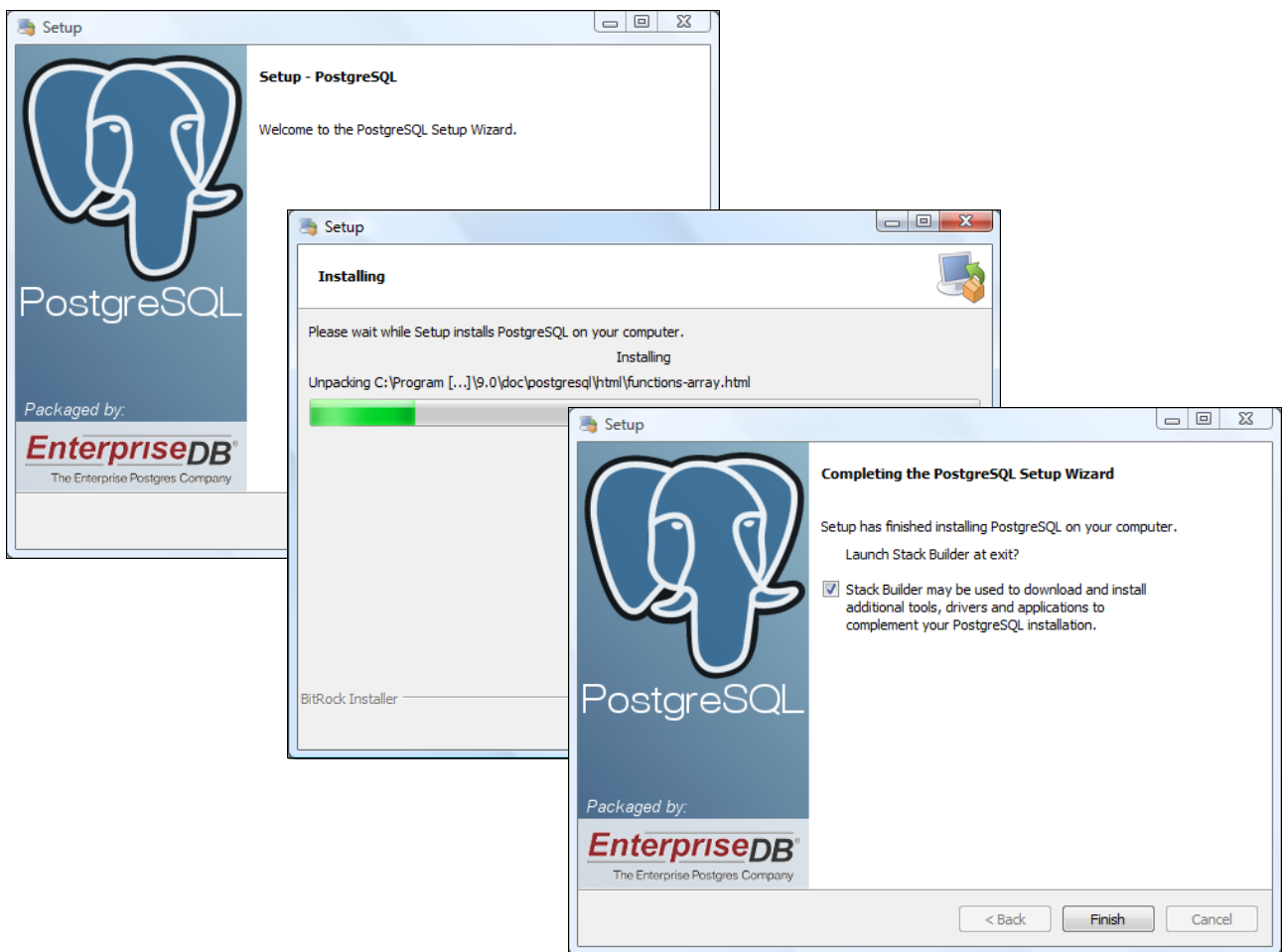
Per realitzar el treball obtenim la darrera versió del programari, versió 9.0.3-1, des del web <http://www.postgresql.org/download/>, que finalment ens redirigeix a <http://www.enterprisedb.com/products-services-training/pgdownload#windows>



Il·lustració 2.2: Pàgina web per descarregar el programari PostgreSQL

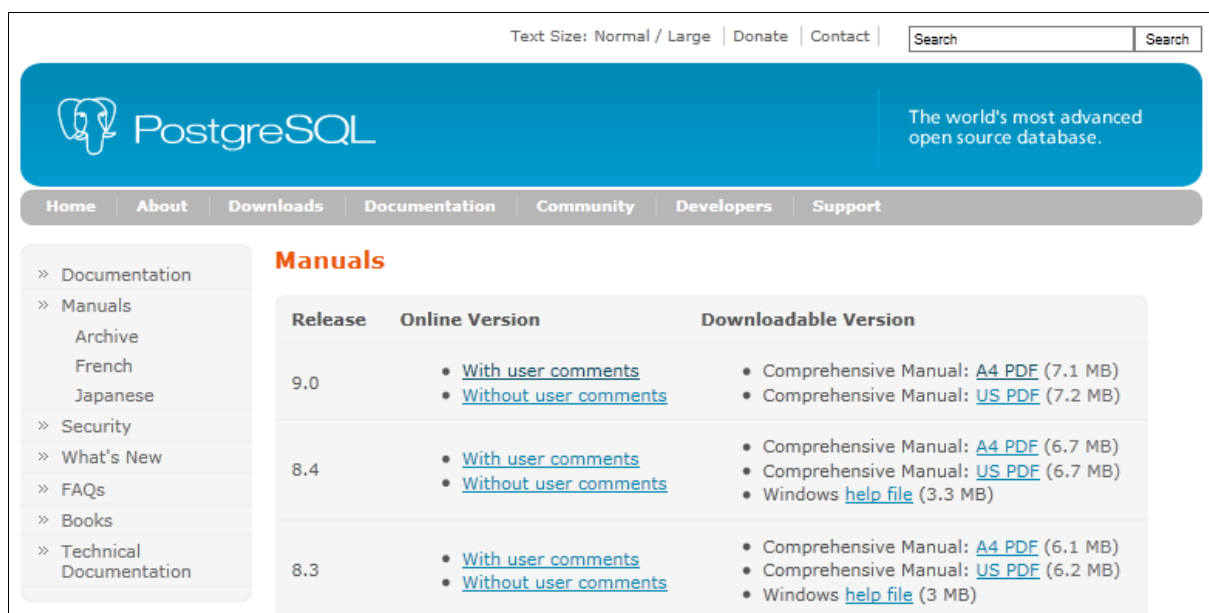
La instal·lació és simple, només cal seguir les indicacions que van apareixent pàgina a pàgina:





Il·lustració 2.3: Procés instal·lació del programari PostgreSQL

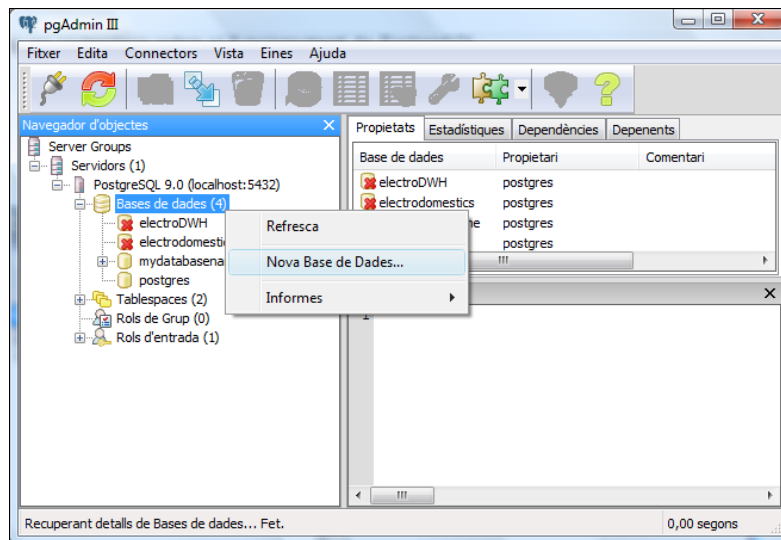
En el web <http://www.postgresql.org/docs/manuals/> hi trobem la documentació:



Il·lustració 2.4: Pàgina web per descarregar documentació PostgreSQL

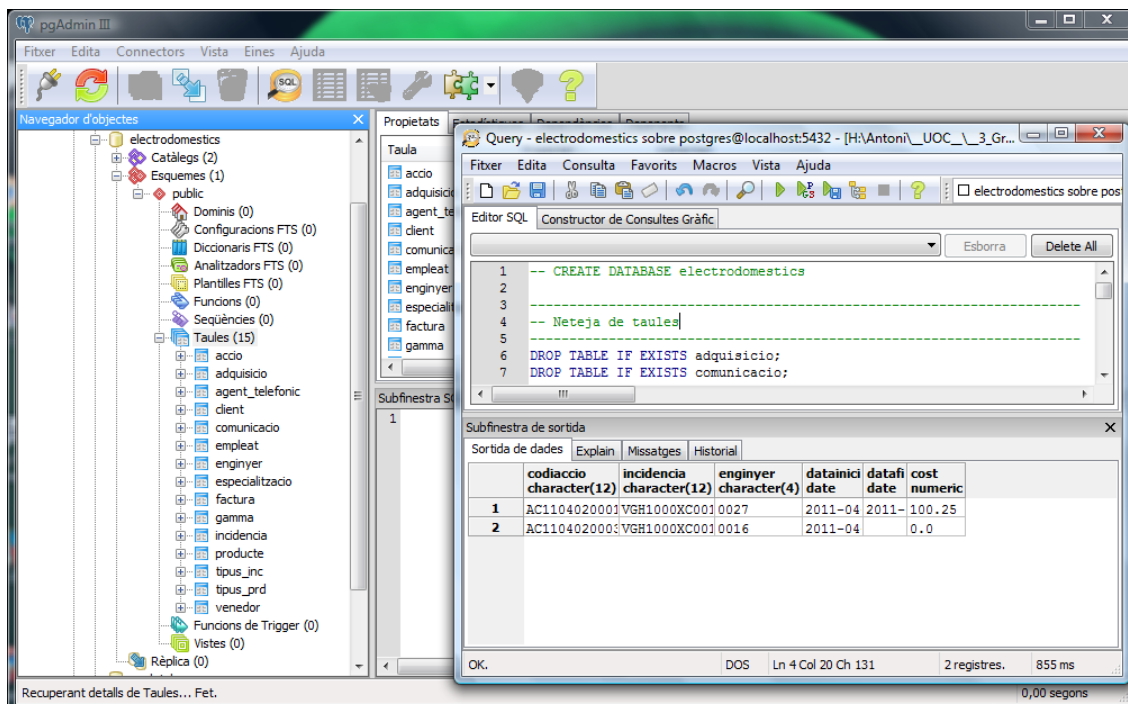
Alguns comentaris sobre el funcionament de PostgreSQL:

- Per crear una base de dades de manera ràpida, ho podem fer per la interfície gràfica de "pgAdmin III", seleccionant en el navegador d'objectes l'apartat "Bases de dades" i amb el botó de la dreta, "Nova Base de Dades...":



Il·lustració 2.5: Crear una base de dades amb "pgAdmin III"

- Per executar un script SQL, primer, en el navegador d'objectes de la interfície gràfica de "pgAdmin III", seleccionem la BD sobre la qual volem executar l'script, i després, prement la icona SQL (executa consultes SQL arbitràries) ens obrirà l'editor SQL des d'on podem introduir el codi SQL o obrir un script i executar-lo:



Il·lustració 2.6: Executar un script SQL amb "pgAdmin III"

## 2.3. Apache Tomcat, servidor web

*Tomcat* és un contenidor de miniaplicacions de servidor (*servlets* en anglès) i de *JavaServer Pages* (JSP), és a dir, un mòdul per executar *servlets* i/o pàgines JSP en aplicacions web, pot funcionar com a servidor d'aplicacions, i l'utilitzarem com a servidor web local. Funciona en qualsevol sistema operatiu que disposi de la màquina virtual *Java*.

Les versions disponibles a la data del treball (març de 2011) són les següents:

Servlet/JSP Spec	Apache Tomcat version	Actual release revision	Minimum Java Version
3.0/2.2	7.0.x	7.0.11	1.6
2.5/2.1	6.0.x	6.0.32	1.5
2.4/2.0	5.5.x	5.5.33	1.4
2.3/1.2	4.1.x (archived)	4.1.40 (archived)	1.3
2.2/1.1	3.3.x (archived)	3.3.2 (archived)	1.1

Il·lustració 2.7: Versions programari Apache Tomcat

Com que per fer el treball s'utilitzarà el Java 1.6, descarreguem la darrera versió, la 7.0.x des de <http://tomcat.apache.org/download-70.cgi>:

**7.0.11**

Please see the [README](#) file for packaging information. It explains what every distribution contains.

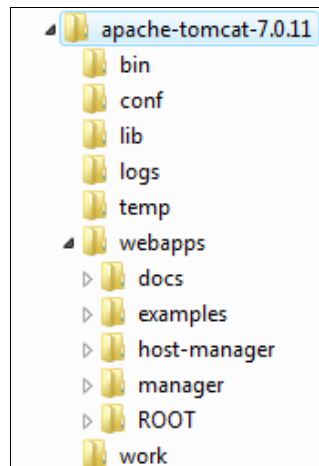
**Binary Distributions**

- Core:
  - [zip \(pgp, md5\)](#)
  - [tar.gz \(pgp, md5\)](#)
  - [32-bit Windows zip \(pgp, md5\)](#)
  - [64-bit Windows zip \(pgp, md5\)](#)
  - [64-bit Itanium Windows zip \(pgp, md5\)](#)
  - [32-bit/64-bit Windows Service Installer \(pgp, md5\)](#)
- Full documentation:
  - [tar.gz \(pgp, md5\)](#)

Il·lustració 2.8: Distribucions Apache Tomcat segons sistema operatiu

Segons el sistema operatiu Windows, descarreguem el fitxer "zip" corresponent; i per instal·lar-lo només cal descomprimir, deixant el programari preparat amb aquesta

estructura de directoris:



Il·lustració 2.9: Estructura de directoris Tomcat

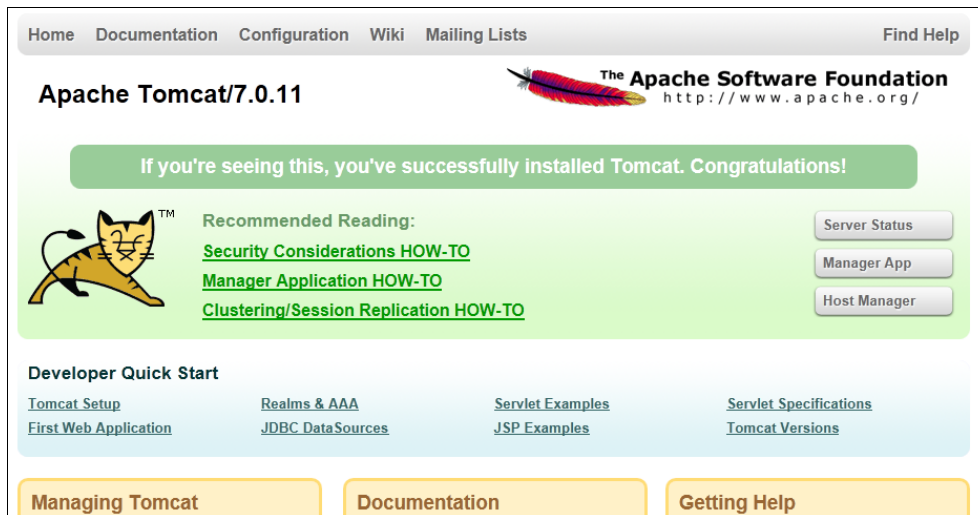
Per comprovar que funciona, només cal fer:

- Engregar el programari, executant el fitxer "*apache-tomcat-7.0.11\bin\startup.bat*". Obrirà una finestra de comandes que romandrà oberta mentre estigui funcionant el programa:

```
25/03/2011 18:56:50 org.apache.catalina.core.AprLifecycleListener init
INFO: Loaded APR based Apache Tomcat Native library 1.1.20.
25/03/2011 18:56:50 org.apache.catalina.core.AprLifecycleListener init
INFO: APR capabilities: IPv6 [true], sendfile [true], accept filters [false], random
[true].
25/03/2011 18:56:51 org.apache.coyote.AbstractProtocolHandler init
INFO: Initializing ProtocolHandler ["http-apr-8080"]
25/03/2011 18:56:51 org.apache.coyote.AbstractProtocolHandler init
INFO: Initializing ProtocolHandler ["ajp-apr-8009"]
25/03/2011 18:56:51 org.apache.catalina.startup.Catalina load
INFO: Initialization processed in 1541 ms
25/03/2011 18:56:51 org.apache.catalina.core.StandardService startInternal
INFO: Starting service Catalina
25/03/2011 18:56:51 org.apache.catalina.core.StandardEngine startInternal
INFO: Starting Servlet Engine: Apache Tomcat/7.0.11
25/03/2011 18:56:51 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory ROOT
25/03/2011 18:56:52 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory docs
25/03/2011 18:56:52 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory examples
25/03/2011 18:56:53 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory host-manager
25/03/2011 18:56:53 org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory manager
25/03/2011 18:56:53 org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ["http-apr-8080"]
25/03/2011 18:56:53 org.apache.coyote.AbstractProtocolHandler start
INFO: Starting ProtocolHandler ["ajp-apr-8009"]
25/03/2011 18:56:53 org.apache.catalina.startup.Catalina start
INFO: Server startup in 1689 ms
```

Il·lustració 2.10: Pantalla d'arrancada del Tomcat

- Amb un navegador accedim a l'adreça URL "<http://localhost:8080/>", i ens mostra una pàgina del *Tomcat* indicant que la instal·lació és correcta:



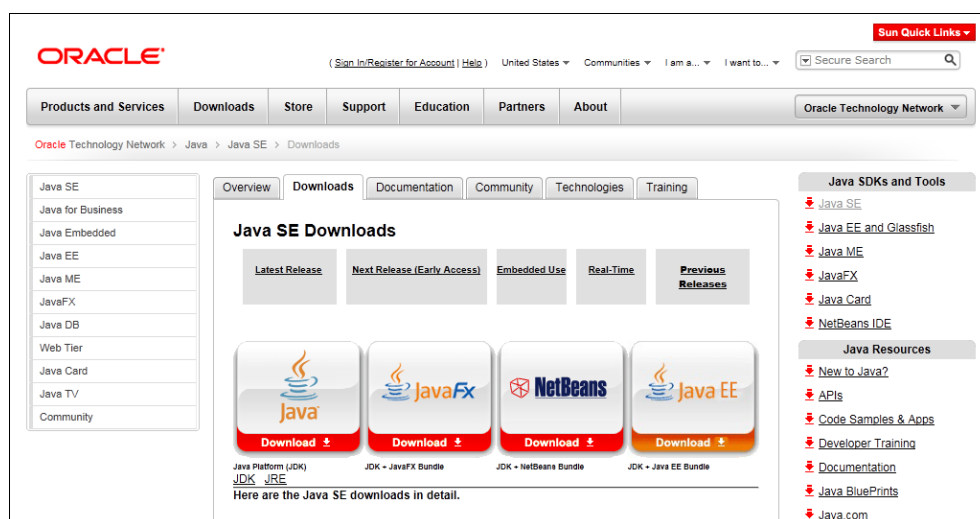
Il·lustració 2.11: Pàgina web de connexió a Apache Tomcat

- Per parar-lo, només cal executar "`apache-tomcat-7.0.11\bin\shutdown.bat`".

## 2.4. Java, llenguatge de programació del servidor web

El *Java* és un llenguatge interpretat, orientat a objectes, flexible, potent, multi-plataforma, i permet treballar amb SQL dinàmic, i és l'indicat per treballar amb *Tomcat*. És el llenguatge de programació que utilitzem en el projecte per interactuar des de les pàgines HTML amb la BD a través del servidor web.

S'ha utilitzat la versió 1.6.0.24, la darrera versió disponible en fer el treball en el web <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.



Il·lustració 2.12: Pàgina web per descàrrega el programari Java

Després d'instal·lar el programari validarem que les següents variables d'entorn continguin els valors correctes:

Variable	Valor
JAVA_HOME	C:\Program Files\Java\jdk1.6.0_24
Path	[...];C:\Program Files\Java\jdk1.6.0_24\bin;[...]

Taula 2.1: Variables d'entorn necessàries pel programari Java

Per accedir a *PostgreSQL* des de *Tomcat* amb *Java* necessitarem el programa controlador (o *driver*) JDBC. Aquest programa controlador ens facilita *PostgreSQL* des de la pàgina web <http://jdbc.postgresql.org/download.html>. Com que utilitzem la versió 1.6 de *Java*, descarreguem el JDBC4, tal i com ens indiquen:



Il·lustració 2.13: Pàgina web per descarregar el JDBC

El fitxer obtingut s'anomena "postgresql-9.0-801.jdbc4.jar", i l'hem de guardar sense modificar en el directori "apache-tomcat-7.0.11\lib\" del *Tomcat*.

## 2.5. Prova del funcionament entre el servidor web i el SGBD

Un cop tenim instal·lats els programaris bàsics: *Java*, *PostgreSQL* i *Tomcat*; i per provar el funcionament correcte, procedim a:

### 2.5.1. Creació d'una BD a PostgreSQL

- Accedim a *PostgreSQL*, (Inicia → Tots els programes → PostgreSQL 9.0 → pgAdmin III).
- Ens connectem a la BD que s'ha creat en la instal·lació "PostgreSQL 9.0 (localhost:5432)".
- Des de la finestra SQL executem el codi següent<sup>2</sup>:

```
CREATE DATABASE mydatabasename
CREATE TABLE employee(name VARCHAR(20), dept VARCHAR(20), jobTitle
VARCHAR(20));
INSERT INTO employee VALUES ('Fred Flinstone','Quarry Worker','Rock Digger');
INSERT INTO employee VALUES ('Wilma Flinstone','Finance','Analyst');
INSERT INTO employee VALUES ('Barney Rubble','Sales','Neighbor');
INSERT INTO employee VALUES ('Betty Rubble','IT','Neighbor');
SELECT * FROM employee;
```

*Il·lustració 2.14: Codi SQL creació de la base de dades mydatabase*

- I obtenim un detall dels registres de la BD:

	<b>name</b> character varying(20)	<b>dept</b> character varying(20)	<b>jobtitle</b> character varying(20)
<b>1</b>	Fred Flinstone	Quarry Worker	Rock Digger
<b>2</b>	Wilma Flinstone	Finance	Analyst
<b>3</b>	Barney Rubble	Sales	Neighbor
<b>4</b>	Betty Rubble	IT	Neighbor

*Il·lustració 2.15: Resultat consulta SQL de la base de dades mydatabase*

### 2.5.2. Preparació servidor Tomcat

- Al directori "apache-tomcat-7.0.11\webapps" del *Tomcat*, creem el directori per a la prova de nom "UOC-TFG-prova", i dins d'aquest nou directori, el directori "WEB-INF".
- Dins del directori "WEB-INF" creem els directoris "classes" i "lib", i el fitxer "web.xml" amb el següent contingut:

<sup>2</sup> Exemple extret del tutorial de Greg Ippolito a <http://www.yolinux.com/>

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0"
  metadata-complete="true">
  <description>
    Servlet and JSP Examples.
  </description>
  <display-name>Servlet and JSP Examples</display-name>
  <servlet>
    <servlet-name>MostraBD</servlet-name>
    <servlet-class>MostraBD</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>MostraBD</servlet-name>
    <url-pattern>/MostraBD</url-pattern>
  </servlet-mapping>
</web-app>

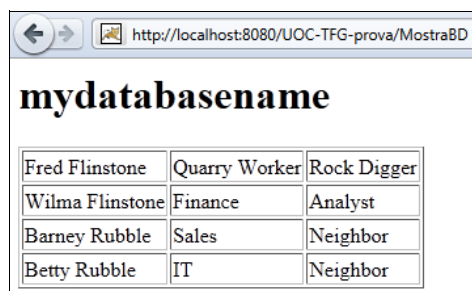
```

Il·lustració 2.16: Codi web.xml per prova de funcionament del Tomcat

- Revisarem que tenim el *driver* JDBC en el directori "apache-tomcat-7.0.11\lib", segons hem indicat a l'apartat 2.4 "Java, llenguatge de programació del servidor web".
- Dins del directori "apache-tomcat-7.0.11\classes" hi posem el programa Java "MostraBD.java" (veure el codi a l'annex 2).
- Compilem el programa per crear "MostraBD.class" dins el mateix directori, assegurant accedir amb el *classpath* a "apache-tomcat-7.0.11\lib\servlet-api.jar".

### 2.5.3. Execució i resultat

- Enguegem el Tomcat (veure apartat 2.3 "Apache Tomcat, servidor web")
- Ens connectem a l'adreça "http://localhost:8080/UOC-TFG-prova/MostraBD"
- Hem d'obtenir el resultat que ens confirma el bon funcionament:



mydatabasename		
Fred Flinstone	Quarry Worker	Rock Digger
Wilma Flinstone	Finance	Analyst
Barney Rubble	Sales	Neighbor
Betty Rubble	IT	Neighbor

Il·lustració 2.17: Resultat prova de consulta al servidor web



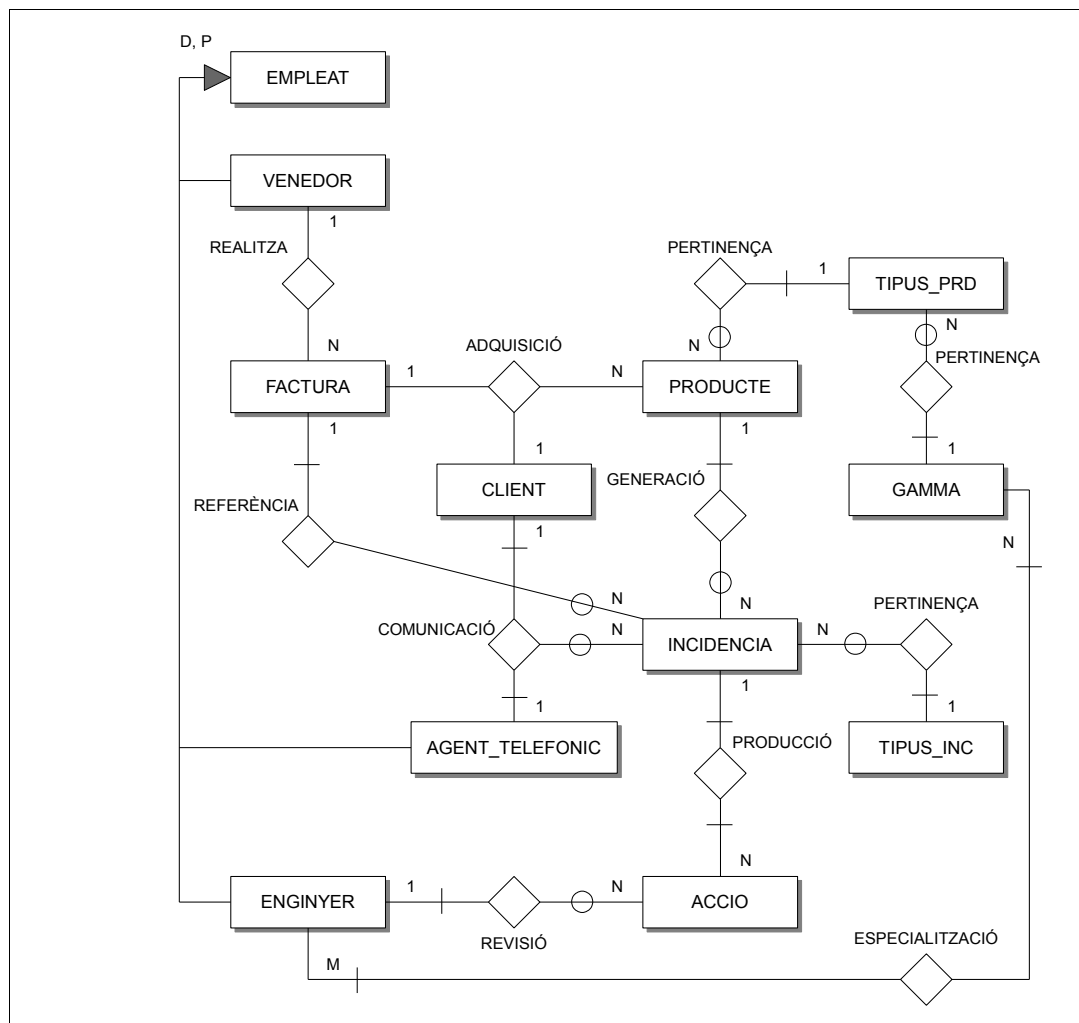
### 3. Base de dades operacional

En aquest capítol s'ha fet l'anàlisi de requeriments, dissenyat i implementat la base de dades per satisfer les necessitats que ens demanen, obtenint com a resultat la BD operacional. A part de la creació de la BD, també es detalla la creació dels procediments emmagatzemats per la gestió, i el sistema de diari de les accions fetes amb la BD que permetrà resoldre problemes potencials que puguin sorgir.

#### 3.1. Creació de la BD

##### 3.1.1. Disseny de la BD

Segons l'anàlisi dels requeriments, el disseny conceptual de la BD que es fa és el següent:



Il·lustració 3.1: Disseny conceptual de la BD

Tenint en compte que el disseny reflecteix que l'empresa per a la qual dissenyem el programari és fabricant d'electrodomèstics, s'ha establert que els clients seran les botigues i cadenes d'electrodomèstics, o sigui, entitats jurídiques. Aquests clients seran els intermediaris entre el client final i l'empresa fabricant, i l'empresa no es relacionarà amb el client final.

Entre els empleats que intervenen en el projecte s'identifiquen tres subgrups o subclasses, que són: venedors, agents telefònics i enginyers. Considerant que aquesta especialització és disjunta (un empleat no pot aparèixer a dues subclasses), i parcial (no tots els empleats de l'empresa han de pertànyer a una d'aquestes subclasses, ja que aquest model només es dissenya per a la gestió d'incidències).

Els productes que es fabriquen es classifiquen per tipus i gammes. Així es preveu que inicialment hi haurà 3 gammes d'electrodomèstics (de cuina, petit, i gran). Així, d'electrodomèstics de cuina hi haurà tipus com: batedora, robot de cuina, cafetera, fregidora... ; electrodomèstics petits com: aspirador, planxa, màquina de vapor, robot de neteja, microones, ventilador, calefactor... ; i electrodomèstics grans com nevera, rentaplats, rentadora, assecadora, congelador... Els enginyers que gestionaran les avaries estaran especialitzats en una gamma, o més d'una, dels productes.

Tindrem un registre de les factures emeses, per tenir registrat els productes venuts, els clients que els han comprat, i els venedors, i així poder saber si el producte adquirit pel client encara està en període de garantia o no.

Les incidències les comunicaran els clients a través d'un agent telefònic, i en fer l'alta d'una incidència es registrarà el tipus. D'inici es preveuen 3 tipus d'incidències: mecàniques, elèctriques i electròniques. Dels tipus d'incidències es diferenciaran les que poden entrar en garantia, i les que no. Així, si una avaria es deguda a una causa provocada pel client final, aquesta no entrarà en garantia. Quan arribi el producte per reparar, l'enginyer iniciarà una acció, i depenent de la reparació es podran obrir més accions fins a solucionar o tancar la incidència.

Inicialment la BD es dissenya amb uns atributs mínims, i segons es vagi desenvolupant l'aplicació, i si es troba necessari, s'ampliaran si és el cas, ja que el sistema és escalable i permet incorporar progressivament totes les necessitats que sorgeixin. Així mateix, s'agrupen algunes dades, com a l'adreça dels clients i empleats, que es podria desglossar en diferents camps, però no és objectiu d'aquest treball.

Els atributs que figuren al diagrama són els següent (les claus primàries s'han subratllat):

EMPLEAT	<u>codiEmpleat</u> , nif, nom, adreça, telefon, nss
VENEDOR (subclasse d'empleat)	<u>codiEmpleat</u> , comissioVendes
AGENT_TELEFONIC (subclasse d'empleat)	<u>codiEmpleat</u> , telefonDepartament
ENGINYER (subclasse d'empleat)	<u>codiEmpleat</u> , especialitat
CLIENT	<u>codiClient</u> , nif, nom, adreça, telefon, descompte
PRODUCTE	<u>codiProducte</u> , descripcio, tipus, preuCost, preuVenda, duradaGarantia
TIPUS_PRD	<u>codiTipusPrd</u> , gamma, descripcio
GAMMA	<u>codiGamma</u> , descripcio
FACTURA	<u>codiFactura</u> , venedor, data
INCIDENCIA	<u>codiIncidencia</u> , producte, factura, tipusIncidencia, dataObertura, dataTancament
TIPUS_INC	<u>codiTipusInc</u> , descripcio, garantiaAplicable?
ACCIO	<u>codiAccio</u> , incidencia, enginyer, dataInici, dataFi, cost

*Il·lustració 3.2: Atributs del disseny conceptual*

El resultat de la transformació a relacional del model ER consta de les següents relacions:

EMPLEAT( <u>codiEmpleat</u> , nif, nom, adreça, telefon, nss)
VENEDOR( <u>codiEmpleat</u> , comissioVendes)
on {codiEmpleat} referencia EMPLEAT

AGENT-TELEFÒNIC(codiEmpleat, telefonDepartament)  
 on {codiEmpleat} referencia EMPLEAT

ENGINYER (codiEmpleat, especialitat)  
 on {codiEmpleat} referencia EMPLEAT

CLIENT(codiClient, nif, nom, adreça, telefon, descompte)

PRODUCTE(codiProducte, descripcio, tipus, preuCost, preuVenda, duradaGarantia)  
 on {tipus} referencia TIPUS\_PRD

TIPUS\_PRD(codiTipusPrd, gamma, descripcio)  
 on {gamma} referencia GAMMA

GAMMA(codiGamma, descripcio)

FACTURA(codiFactura, venedor, data)  
 on {venedor} referencia VENEDOR

INCIDENCIA(codiIncidencia, descripcio, producte, factura, tipusIncidencia, dataObertura,  
 dataTancament)  
 on {producte} referencia PRODUCTE  
 on {factura} referencia FACTURA  
 on {tipusIncidencia} referencia TIPUS\_INC

TIPUS\_INC(codiTipusInc, descripció, garantiaAplicable?)

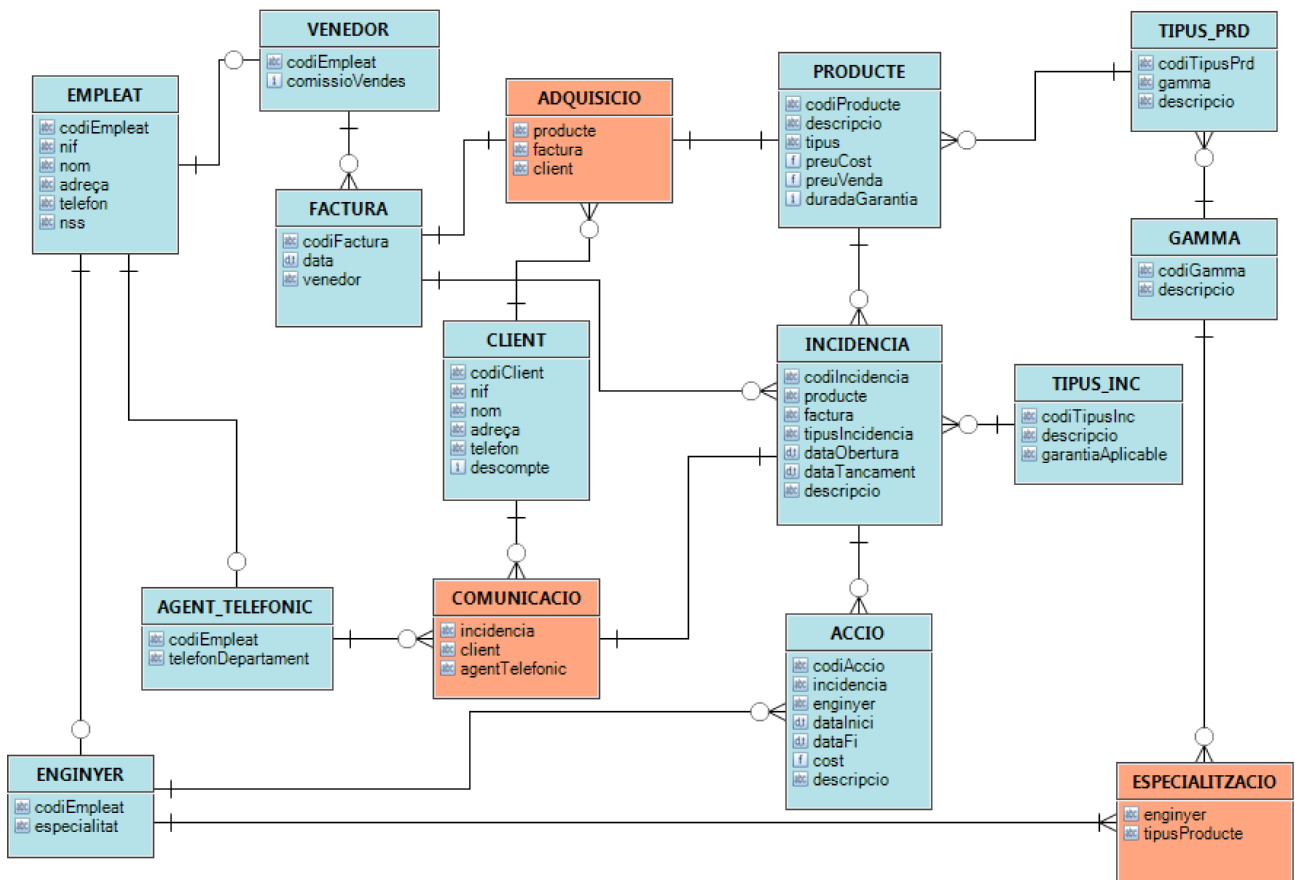
ACCIO(codiAccio, incidencia, enginyer, descripcio, dataInici, dataFi, cost)  
 on {incidencia} referencia INCIDENCIA  
 on {enginyer} referencia ENGINYER

ESPECIALITZACIO(enginyer, gamma)  
 on {enginyer} referencia ENGINYER  
 on {gamma} referencia GAMMA

ADQUISICIO(producte, factura, client)  
 on {producte} referencia PRODUCTE  
 on {factura} referencia FACTURA  
 on {client} referencia CLIENT

COMUNICACIO(incidencia, client, agentTelefonic)  
 on {incidencia} referencia a INCIDENCIA  
 on {client} referencia a CLIENT  
 on {agentTelefonic} referencia a AGENT\_TELEFONIC

Il·lustració 3.3: Relacions del model ER



Il·lustració 3.4: Diagrama ER de la BD

### 3.1.2. Scripts de creació de la BD

A partir del model ER procedim a implementar la base de dades, creant la base de dades i les taules. Primer creem la base de dades:

```
CREATE DATABASE electrodomestics
```

Il·lustració 3.5: Codi SQL de la creació de la BD "electrodomestics"

I tot seguit les taules. Per crear les taules utilitzem un script anomenat "agarrigarov\_TFG\_script\_1-taules-BD.sql". Podem observar la part bàsica d'aquest script en l'annex 3.

### 3.2. Sistema de diari (*log*)

S'ha implementat el sistema de diari (*log*) detallat de les accions fetes a la BD per poder detectar problemes que puguin aparèixer. Aquest sistema es basa en la creació d'una taula, anomenada 'log' (ref. "agarrigarov\_TFG\_script\_2-log-BD.sql"):

```

-----
-- Taula log -- Log de les operacions que es fan a la BD
-----
DROP TABLE IF EXISTS log;
CREATE TABLE log(
  numero SERIAL CONSTRAINT PK_log PRIMARY KEY,
  data DATE CONSTRAINT NN_log_data NOT NULL,
  hora TIME CONSTRAINT NN_log_hora NOT NULL,
  usuari VARCHAR(32) CONSTRAINT NN_log_usuari NOT NULL,
  operacio CHAR(6) CONSTRAINT NN_log_operacio NOT NULL,
  taula VARCHAR(16) CONSTRAINT NN_log_taula NOT NULL,
  CONSTRAINT CH_log_inc_gar
  CHECK(operacio IN ('INSERT','UPDATE','DELETE')),
  dades VARCHAR(512) CONSTRAINT NN_log_operacio NOT NULL
);

```

Il·lustració 3.6: Codi SQL de la creació de la taula 'log'

S'ha creat una funció anomenada 'gravar\_log' per centralitzar les altes a la taula, que es cridarà des dels procediments (ref. "agarrigarov\_TFG\_script\_3-proc-BD.sql"):

```

CREATE OR REPLACE FUNCTION gravar_log(o NAME, t TEXT, dd TEXT) RETURNS void AS $$
BEGIN
  INSERT INTO log (data, hora, usuari, operacio, taula, dades)
  VALUES (current_date, clock_timestamp(), current_user, $1, $2, $3);
END;
$$ LANGUAGE plpgsql;

```

Il·lustració 3.7: Funció 'gravar\_log' per centralitzar gravacions al diari

Només caldrà cridar a la funció en el següent format:

```
"var = gravar_log(operacio, taula, dades);"
```

On a "operacio" hi posarem un dels valors previstos: *INSERT*, *UPDATE* o *DELETE*, a "taula" hi posarem la taula on es fa l'operació, i a "dades" la informació que ens interessi guardar, i que sigui explicativa de l'operació a realitzar. A tall d'exemple:

	numero [PK] serial	data date	hora time without	usuari character var	operacio character(6)	taula character var	dades character varying(512)
1	7068	2011-04-16	00:34:31.22	postgres	INSERT	log	Inici del log de la BD Electrodomestics
2	7069	2011-04-16	00:34:31.24	postgres	INSERT	gamma	C:Cuina;

Il·lustració 3.8: Registres gravats a la taula 'log'

### 3.3. Procediments emmagatzemats

Un cop definides les taules es creen els procediments emmagatzemats que es dispararan automàticament en les actualitzacions de les dades, el que s'anomena disparadors o *triggers* amb anglès, el que farà que la BD es mantingui íntegra quan hi hagi sentències de manipulació de BD (*INSERT*, *UPDATE*, *DELETE*).

Es crea un procediment per a cada taula, que gestionarà les sentències de manipulació de la BD. En cada procediment, primer es controlen els valors obligatoris i les interrelacions entre les taules, tot i que el SGBD ho fa a partir de la definició de les taules, es fa per retornar un missatge més explícit a l'usuari. Després, i si és el cas, es controlen altres restriccions que no es poden posar en la creació de les taules. I finalment, es grava l'acció realitzada en el diari cridant a la funció 'gravar\_log' (veure apartat 3.2 "Sistema de diari (*log*)").

L'script preparat s'anomena "*agarrigarov\_TFG\_script\_3-proc-BD.sql*", i a tall d'exemple s'adjunta en l'annex 4 el procediment de la taula 'accio', on també es poden observar els controls interrelacionals per validar les dates entre l'acció i la incidència.

### 3.4. Prova de càrrega de dades a la BD

Per testejar les funcionalitats de la BD es crea l'script "*agarrigarov\_TFG\_script\_4-dades-BD.sql*", que com la resta d'scripts, s'adjunta a aquest document. Executant aquest script ens carregarà una sèrie de valors a la BD.

	codiproducte [PK] character(12)	descripcio character varying(32)	tipus character(4)	preucost numeric	preuvenda numeric	duradagarantia integer
1	B3VS3BP00104	PLAT-NET 3VS300BP	0002	160.49	329.0	4
2	E44500X00009	NEV-NEW END44500X	0001	366.50	759.0	2
3	EDL124700056	ROBA-NETA ROMAN-L1247 7KG	0003	197.85	419.0	4
4	IRRB53100042	ROBOT-NET ROOMBA531	0009	145.75	369.0	3
5	LG1929G00199	ONES-ELUX MS 1929G	0010	33.25	65.0	2
6	MLJ204500007	LIQUS-FRUIT VITAE JU 200045	0014	22.0	44.0	2
7	OTV670F00045	ROBA-SECA VOHV670F	0004	147.70	369.0	3
8	SC100MC00012	ROBOT-XEF CF100, MULTIICOOKER	0015	74.99	149.0	2
9	SMS884500062	ROBOT-ELUX SR8845 Navibot	0009	142.35	329.0	3
10	TAP220000077	FERRO-ROENT PTSI 2200	0007	8.75	16.70	1

Il·lustració 3.9: Consulta de productes a la BD

## 4. Magatzem de dades

En aquest capítol s'ha fet l'anàlisi de requeriments, dissenyat i implementat el magatzem de dades (*datawarehouse* o DWH), així com els processos ETL (extracció, transformació i càrrega) pel traspàs d'informació entre la base de dades operacional i el DWH.

### 4.1. Creació del magatzem de dades

#### 4.1.1. Requisits

Després de valorar els requeriments inicials de l'empresa sobre com pot millorar l'eficiència a partir de la gestió diària de les incidències a través de definir un magatzem de dades, inicialment es defineix la necessitat de consolidar les dades referents a les incidències dels productes i el cost de les garanties, per més endavant poder obtenir estadístiques i informació d'ajuda per a la millora. Així, doncs, s'han definit els següents informes que es volen obtenir de les dades del magatzem que hem de construir:

- Nombre d'incidències per tipus de producte i/o tipus d'incidència.
- Cost de les garanties per tipus de producte i/o tipus d'incidència.
- Cost de les garanties en un determinat període.
- Cost de les incidències en un determinat període.

Més endavant, i al ser un sistema escalable, hi haurà la possibilitat d'ampliar els informes a realitzar.

#### 4.1.2. Anàlisi i disseny

Per dissenyar el magatzem de dades, apliquem el model conceptual multidimensional<sup>3</sup>, que està marcat per la dicotomia fet-dimensió, que distingeix entre els dos tipus de dades: els fets que volem analitzar i les dimensions que utilitzem per a analitzar-los. Un fet representa un tema objecte d'anàlisi, i una dimensió representa un punt de vista que utilitzarem en l'anàlisi de les dades. Un fet i el seu corresponent conjunt de dimensions formen una estrella. A partir del model conceptual obtindrem el model físic.

---

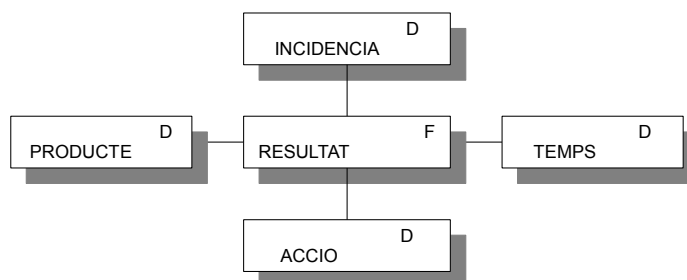
<sup>3</sup> Abelló Gamazo, Alberto. *Disseny multidimensional*.



### - Model conceptual

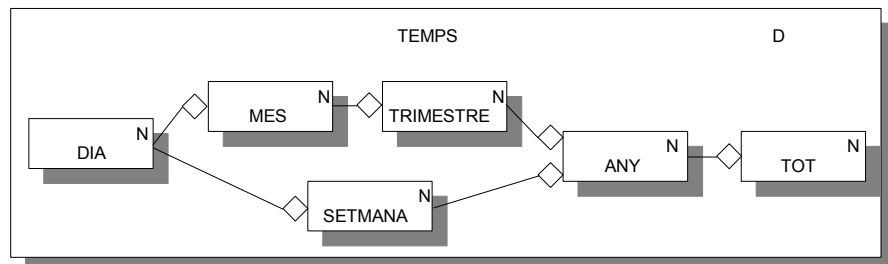
A partir de les dades de que disposem de la base de dades operacional, i segons els informes desitjats, podem establir quins fets, o temes d'anàlisi, hem de tenir presents en el model.

Aplicant el mètode per dissenyar l'estrella bàsica, obtenim l'estrella que ens permetrà obtenir els informes a partir d'un fet, el resultat, que tant ens representarà el nombre d'incidències o el seu cost:



Il·lustració 4.1: Estrella bàsica

Amb la dimensió Temps podrem obtenir diferents resultats ajustant el nivell de la dimensió:



Il·lustració 4.2: Dimensió Temps

Tot seguit es detallen els atributs que pertanyen a les altres dimensions i que creiem que ens seran útils per a triar (seleccionar o agrupar) o descriure l'espai d'anàlisi:

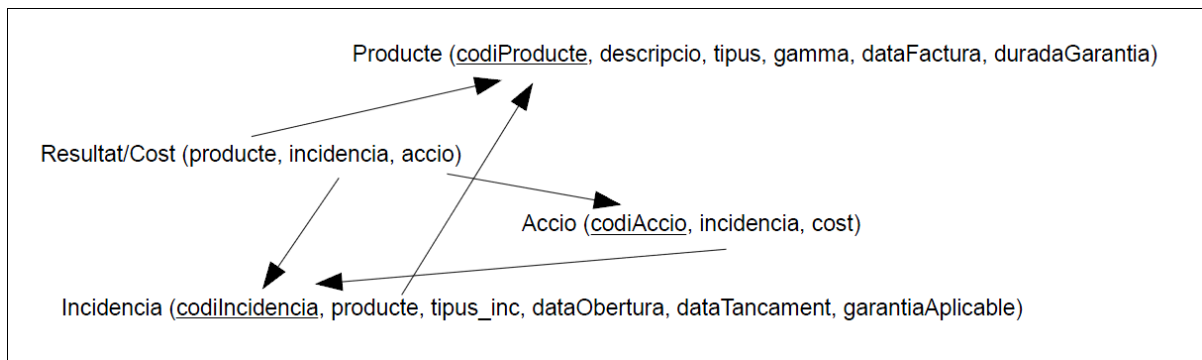
Dimensió	Atributs
Producte	codiProducte, descripcio, tipus, gamma, dataFactura, duradaGarantia
Incidència	Base de dades operacionalcodiIncidencia, producte, tipus_inc, dataObertura, dataTancament, garantiaAplicable
Accio	codiAccio, incidencia, cost

Taula 4.1: Atributs de les dimensions model conceptual DWH

El fet resultat no tindrà mesures o atributs en les cel·les, ja que les dades necessàries les tindrem en les dimensions.

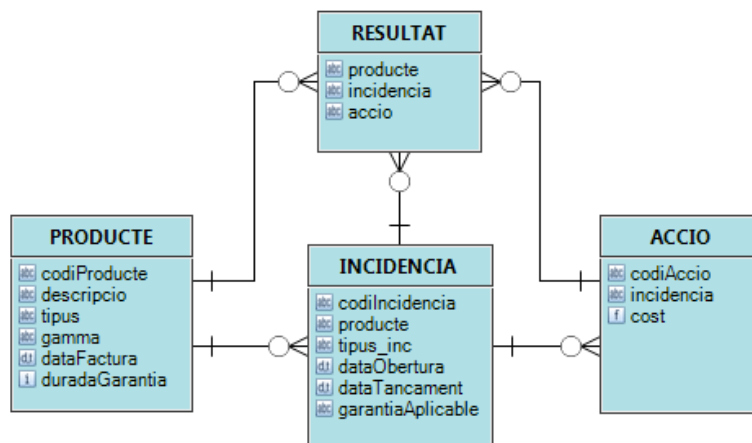
### - Model físic

A partir dels models conceptuals passem al model físic, obtenint l'esquema relacional amb forma d'estrella. El fet, tal i com hem indicat, no conté mesures perquè amb els atributs de les dimensions ja tindrem les dades que necessitem, per tant, només contindrà les claus foranes cap a les dimensions.



Il·lustració 4.3: Esquema relacional

Fem el disseny lògic valorant l'estrella. Per implementar-ho creem una taula pel fet i una taula per a cada dimensió. La taula del fet està lligada per claus foranes amb les taules de les dimensions.



Il·lustració 4.4: Diagrama ER del DWH

### 4.1.3. Implementació

Implementem el disseny anterior creant la base de dades 'electroDWH' i les taules, tenint en compte que el magatzem de dades s'omplirà a partir de la BD operacional, no controlarem totes les restriccions, com les dates, ja que donem com a precondició vàlida que les dades ja venen validades.

Cal indicar que hem optat per canviar el nom de les taules implementades i els noms dels atributs del disseny per facilitar la codificació. Així, per exemple, la dimensió 'producte' es representarà per la taula 'DWH\_prd', i l'atribut 'duradaGarantia' per 'DWH\_prd\_tmGar'.

Per crear les taules utilitzem un script anomenat "*agarrigarov\_TFG\_script\_5-taules-DWH.sql*" que podem observar en l'annex 5.

Com que la taula de fets 'DWH\_res' no conté atributs que no siguin de les dimensions, es prepara un procediment que al carregar o actualitzar la taula d'accions, anomenada 'DWH\_acc', ja ens carregui automàticament la taula. Podem veure'n el codi en l'annex 6, que correspon a l'script anomenat "*agarrigarov\_TFG\_script\_6-proc-DWH.sql*".

#### 4.1.4. Proves

Per comprovar el magatzem de dades, executem un joc de proves per comprovar el funcionament, que consisteix en la inserció de diferents registres a les taules, provocant errors controlats i revisar el comportament de les restriccions. Per realitzar-ho s'ha creat i utilitzat l'script "*agarrigarov\_TFG\_script\_7-proves-DWH.sql*".

## 4.2. Processos ETL (extracció, transformació i càrrega)

Un cop ja tenim el magatzem de dades, preparem els processos d'ETL. Extraurem les dades de la base de dades operacional, les transformarem, i tot seguit les carreguem al DWH.

### 4.2.1. Extracció de dades de la BD operacional

Per extreure les dades utilitzem el programa '*SQL shell (psql)*' del programari del *PostgreSQL*. Primerament executem el programa i ens connectem a la base de dades operacional:

```
Server [localhost]:
Database [postgres]: electrodomestics
Port [5432]:
Username [postgres]:
psql (9.0.3)
```

```

WARNING: Console code page (850) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.

electrodomestics=# help
You are using psql, the command-line interface to PostgreSQL.
Type: \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
electrodomestics=#

```

I comencem per extreure les dades que carregarem a la taula 'DWH\_acc'. Primer indiquem en quin fitxer volem gravar les dades, i que només volem els registres, sense capçaleres:

```

Electrodomestics=# \o C:/Users/Antoni/Documents/agarrigarov_TFG_import_DW_acc.csv
electrodomestics=# \t on
Showing only tuples.
electrodomestics=#

```

Tot seguit executem la comanda per extreure la informació:

```

electrodomestics=# select ' ' || a.codiaAccio || ';' || a.incidencia || ';' || a.cost
electrodomestics-#   from accio a, incidencia i
electrodomestics-#   where a.incidencia = i.codiaIncidencia
electrodomestics-#         and i.dataTancament IS NOT NULL;
electrodomestics=#

```

Que obtenim en el fitxer amb el format esperat:

```

electrodomestics=# \! type C:\Users\Antoni\Documents\agarrigarov_TFG_import_DW_acc.csv
ACC2011V0001;INC2011X0004;15.00
ACC2011V0002;INC2011X0004;32.00
ACC2011V0004;INC2011X0005;100.25
ACC2011V0009;INC2011X0005;15.50
ACC2011V0013;INC2011X0008;15.65
ACC2011V0024;INC2011X0009;7.95
ACC2011V0027;INC2011X0014;16.88
ACC2011V0033;INC2011X0019;12.00
ACC2011V0034;INC2011X0019;67.50
ACC2011V0037;INC2011X0025;27.50

```

Repetim el procés per cada taula que volem extreure:

```

electrodomestics=# \o C:/Users/Antoni/Documents/agarrigarov_TFG_import_DW_inc.csv
electrodomestics=# select i.codiaIncidencia || ';' || i.producte || ';' || t.descripcion
electrodomestics-#   from incidencia i left outer join tipus_inc t

```

```

electrodomestics=# on t.codiTipesInc = i.tipusIncidencia
electrodomestics=# where i.dataTancament IS NOT NULL;
electrodomestics=# \! type C:\Users\Antoni\Documents\agarrigarov_TFG_import_DW_inc.csv
INC2011X0004;LG1929G00199;Electronica;2011-03-29;2011-04-12;S
INC2011X0005;ZNZ332W00024;Electrica;2011-04-04;2011-04-11;S
INC2011X0008;TAT4F2V02109;Mecanica;2011-04-06;2011-04-08;S
INC2011X0009;UFCG72200004;Mecanica;2011-05-02;2011-05-12;N
INC2011X0014;ZNZ332W00025;Electrica;2011-05-02;2011-05-16;S
INC2011X0019;B3VS3BP00104;Mecanica;2011-05-09;2011-05-11;N
INC2011X0025;E44500X00009;Mecanica;2011-04-21;2011-04-27;S

electrodomestics=# \o C:/Users/Antoni/Documents/agarrigarov_TFG_import_DW_prd.csv
electrodomestics=# select i.producte ||';'|| p.descripcion ||';'|| t.descripcion ||';'||
electrodomestics=# g.descripcion ||';'|| f.data ||';'|| p.duradaGarantia
electrodomestics=# from (incidencia i left outer join factura f on (i.factura =
electrodomestics=# f.codifactura)),
electrodomestics=# (incidencia ii left outer join
electrodomestics=# (producte p left outer join
electrodomestics=# (tipus_prd t left outer join gamma g on (t.gamma =
electrodomestics=# g.codigamma))
electrodomestics=# on (p.tipus = t.coditipesPrd))
electrodomestics=# on (ii.producte = p.codiproducte))
electrodomestics=# where i.producte = ii.producte
electrodomestics=# and i.dataTancament IS NOT NULL;
electrodomestics=# \! type C:\Users\Antoni\Documents\agarrigarov_TFG_import_DW_prd.csv
LG1929G00199;ONES-ELUX MS 1929G;Microones;Petit;2006-09-27;2
ZNZ332W00024;NEV-ZRT 332W;Nevera;Gran;2010-09-27;2
TAT4F2V02109;VENT-ADOR Tropica 4F 2V;Ventilador;Petit;2010-10-15;3
UFCG72200004;OLI-ROENT CG 7220;Cafetera;Cuina;2010-10-15;2
ZNZ332W00025;NEV-ZRT 332W;Nevera;Gran;2007-01-05;2
B3VS3BP00104;PLAT-NET 3VS300BP;Rentaplats;Gran;2010-12-12;4
E44500X00009;NEV-NEW END44500X;Nevera;Gran;2011-04-20;2

```

En el fitxer "agarrigarov\_TFG\_script\_8-extrac-BD.psql", que s'adjunta al treball, hi ha guardades les comandes utilitzades per fer l'extracció.

#### 4.2.2. Transformació de les dades

Les dades obtingudes en els fitxers s'han de transformar, ja que sinó no es carregaran correctament. Per evitar els errors, com poden ser errors de longitud de clau, o errors en no trobar dades en un registre on només hi ha un caràcter de fi de registre (o retorn de carro), cal fer a cada fitxer obtingut:

- Eliminar els espais inicials de cada registre.
- Eliminar el darrer registre del fitxer, que correspon a un caràcter de fi de registre.

#### 4.2.3. Càrrega de dades al DWH

Primer cal executar el programa 'SQL shell (psql)' del programari del PostgreSQL per connectar al magatzem de dades.

Per carregar les dades al magatzem de dades, utilitzarem la comanda *psql* 'copy' de *PostgreSQL*. El format de la comanda és la següent:

```
electroDWH=# \h copy
Command:      COPY
Description:  copy data between a file and a table
Syntax:
COPY table_name [ ( column [, ...] ) ]
  FROM { 'filename' | STDIN }
  [ [ WITH ] ( option [, ...] ) ]

COPY { table_name [ ( column [, ...] ) ] | ( query ) }
  TO { 'filename' | STDOUT }
  [ [ WITH ] ( option [, ...] ) ]

where option can be one of:

  FORMAT format_name
  OIDS [ boolean ]
  DELIMITER 'delimiter_character'
  NULL 'null_string'
  HEADER [ boolean ]
  QUOTE 'quote_character'
  ESCAPE 'escape_character'
  FORCE_QUOTE { ( column [, ...] ) | * }
  FORCE_NOT_NULL ( column [, ...] )
```

Però per poder executar la comanda amb els fitxers extrets de la BD operacional on hi tenim les dades, ens cal tenir permisos d'accés per a l'usuari 'postgres', ja que altrament ens donarà l'error següent:

```
electroDWH=# COPY DWH_prd FROM
'c:/users/antoni/documents/agarrigarov_TFG_import_DW_acc.csv' USING DELIMITERS ',';
ERROR:  could not open file
"c:/users/antoni/documents/agarrigarov_TFG_import_DW_prd.csv" for reading: Permission
denied
```

Així que donem accés de lectura a l'usuari des del sistema operatiu:

Nom	Nivell de permís
 Antoni	Propietari ▼
 postgres	Lector ▼

Un cop tenim els permisos i els fitxers preparats, procedim a fer la càrrega executant el següent fitxer de comandes o script anomenat "*agarrigarov\_TFG\_script\_9-carregar-DWH.sql*":

```

=====
TRUNCATE TABLE DWH_res, DWH_acc, DWH_inc, DWH_prd;
=====
COPY DWH_prd FROM 'C:/Users/Antoni/Documents/agarrigarov_TFG_import_DW_prd.csv'
USING DELIMITERS ',';
COPY DWH_inc FROM 'C:/Users/Antoni/Documents/agarrigarov_TFG_import_DW_inc.csv'
USING DELIMITERS ',';
COPY DWH_acc FROM 'C:/Users/Antoni/Documents/agarrigarov_TFG_import_DW_acc.csv'
USING DELIMITERS ',';
=====

```

Si revisem el magatzem de dades, veurem que ja hem carregat els registres correctament al magatzem de dades:

	<b>dwh_res_prod</b> [PK] character(12)	<b>dwh_res_incid</b> [PK] character(12)	<b>dwh_res_accio</b> [PK] character(12)
<b>1</b>	B3VS3BP00104	INC2011X0019	ACC2011V0033
<b>2</b>	B3VS3BP00104	INC2011X0019	ACC2011V0034
<b>3</b>	E44500X00009	INC2011X0025	ACC2011V0037
<b>4</b>	LG1929G00199	INC2011X0004	ACC2011V0001
<b>5</b>	LG1929G00199	INC2011X0004	ACC2011V0002
<b>6</b>	TAT4F2V02109	INC2011X0008	ACC2011V0013
<b>7</b>	UFCG72200004	INC2011X0009	ACC2011V0024
<b>8</b>	ZNZ332W00024	INC2011X0005	ACC2011V0004
<b>9</b>	ZNZ332W00024	INC2011X0005	ACC2011V0009
<b>10</b>	ZNZ332W00025	INC2011X0014	ACC2011V0027
*			

*Il·lustració 4.5: Dades carregades al magatzem de dades*

## 5. Consultes al magatzem de dades

En aquest capítol procedim a obtenir els informes que havíem previst en el disseny inicial del DWH. Després d'analitzar els requeriments i les dades que tenim al magatzem de dades, dissenyem i implementem les següents consultes:

### 5.1. Nombre d'incidències per tipus d'incidència

```
select DWH_inc_tipus as Tipus_incidencia, count(distinct DWH_res_incid) as Nombre
  from DWH_res, DWH_inc
 where DWH_res_incid = DWH_inc_codi
 group by DWH_inc_tipus;
```

	tipus_incidencia character varying(32)	nombre bigint
1	Electrica	2
2	Electronica	1
3	Mecanica	4

Il·lustració 5.1: Consulta 1 al DWH

### 5.2. Nombre d'incidències per tipus de producte

```
select DWH_prd_tipus as Tipus_producte, count(distinct DWH_res_incid) as Nombre
  from DWH_res, DWH_prd
 where DWH_res_prod = DWH_prd_codi
 group by DWH_prd_tipus;
```

	tipus_producte character varying(32)	nombre bigint
1	Cafetera	1
2	Microones	1
3	Nevera	3
4	Rentaplats	1
5	Ventilador	1

Il·lustració 5.2: Consulta 2 al DWH

### 5.3. Cost de les garanties per tipus d'incidència

```
select DWH_inc_tipus as Tipus_incidencia, sum(DWH_acc_cost) as Cost_garantia
  from DWH_res, DWH_inc, DWH_acc, DWH_prd
 where DWH_res_incid = DWH_inc_codi
       and DWH_res_accio = DWH_acc_codi
       and DWH_res_prod = DWH_prd_codi
       and DWH_inc_ingar = 'S'
       and DWH_inc_dtIni <= (DWH_prd_data + (DWH_prd_tmgar * interval '1 year'))
 group by DWH_inc_tipus;
```

	tipus_incidencia character varying(32)	cost_garantia numeric
1	Electrica	115.75
2	Mecanica	43.15

Il·lustració 5.3: Consulta 3 al DWH



## 5.4. Cost de les garanties per tipus de producte

```
select DWH_prd_tipus as Tipus_producte, sum(DWH_acc_cost) as Cost_garantia
  from DWH_res, DWH_inc, DWH_acc, DWH_prd
 where DWH_res_incid = DWH_inc_codi
       and DWH_res_accio = DWH_acc_codi
       and DWH_res_prod = DWH_prd_codi
       and DWH_inc_ingar = 'S'
       and DWH_inc_dtIni <= (DWH_prd_data + (DWH_prd_tmgar * interval '1 year'))
 group by DWH_prd_tipus;
```

	tipus_producte character varying(32)	cost_garantia numeric
1	Nevera	143.25
2	Ventilador	15.65

Il·lustració 5.4: Consulta 4 al DWH

## 5.5. Cost de les incidències en un determinat període

```
select 'De 2010-01-01 a 2011-12-31'::text as periode, sum(DWH_acc_cost) as Cost_incidencies
  from DWH_res, DWH_inc, DWH_acc, DWH_prd
 where DWH_res_incid = DWH_inc_codi
       and DWH_res_accio = DWH_acc_codi
       and DWH_res_prod = DWH_prd_codi
       and DWH_inc_dtIni >= to_date('2011-01-01','YYYY-MM-DD')
       and DWH_inc_dtFi <= to_date('2011-12-31','YYYY-MM-DD');
```

	periode text	cost_incidencies numeric
1	De 2010-01-01 a 2011-12-31	310.23

Il·lustració 5.5: Consulta 5 al DWH

## 5.6. Cost de les garanties en un determinat període

```
select 'De 2010-01-01 a 2011-04-15'::text as periode, sum(DWH_acc_cost) as Cost_garantia
  from DWH_res, DWH_inc, DWH_acc, DWH_prd
 where DWH_res_incid = DWH_inc_codi
       and DWH_res_accio = DWH_acc_codi
       and DWH_res_prod = DWH_prd_codi
       and DWH_inc_ingar = 'S'
       and DWH_inc_dtIni <= (DWH_prd_data + (DWH_prd_tmgar * interval '1 year'))
       and DWH_inc_dtIni >= to_date('2011-01-01','YYYY-MM-DD')
       and DWH_inc_dtFi <= to_date('2011-04-15','YYYY-MM-DD');
```

	periode text	cost_garantia numeric
1	De 2010-01-01 a 2011-04-15	131.40

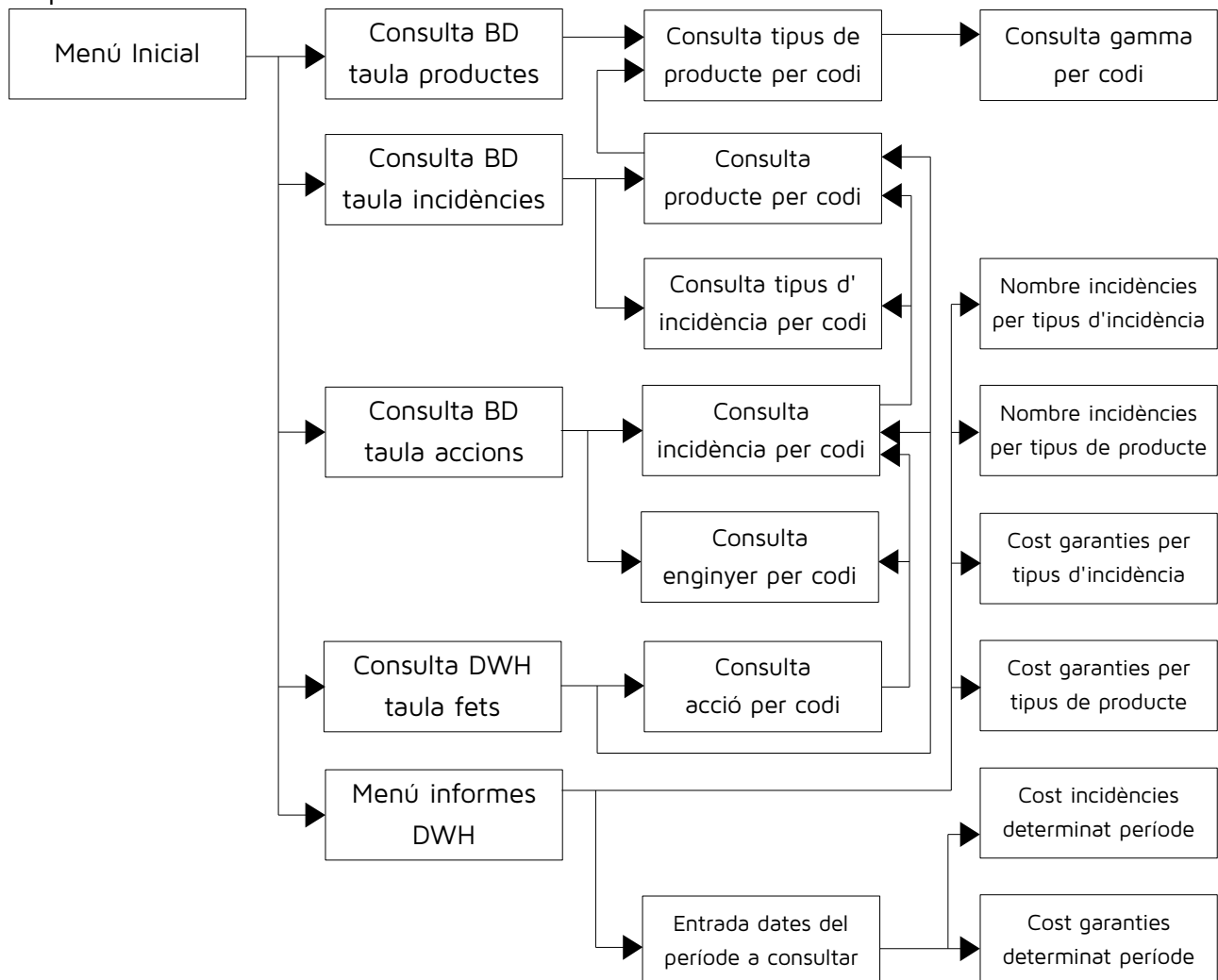
Il·lustració 5.6: Consulta 6 al DWH

## 6. Consultes des d'un navegador web

En aquest capítol es prepara una interfície centralitzada perquè els usuaris puguin realitzar, de manera fàcil i transparent, les consultes al magatzem de dades, i a dades rellevants de la BD, a través d'una pàgina web. El sistema utilitzat és el que hem explicat en el capítol de preparació del programari, punt 2.5, utilitzant el servidor *Apache Tomcat* que ens permet fer les consultes des d'un navegador web.

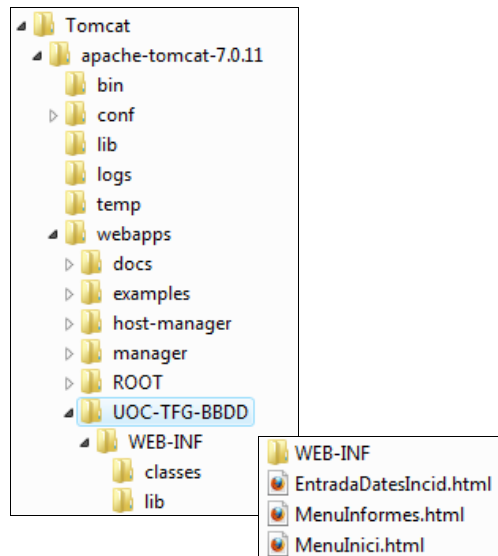
### 6.1. Pàgines web del client

En aquest apartat es dissenya i implementa la part del client, l'entorn web amb què interaccionarà amb la BD i el DWH per fer les consultes. Ens basarem en el següent mapa web:



Il·lustració 6.1: Mapa web

A partir del disseny inicial del mapa web, realitzem amb codi HTML les pàgines: “Menú inicial”, “Menú informes DWH” i “Entrada dates del període a consultar”, i que guardem a la carpeta “UOC-TFG-BBDD” que creem a dins del directori “apache-tomcat-7.0.11\webapps\” on tenim el programari *Apache Tomcat*:



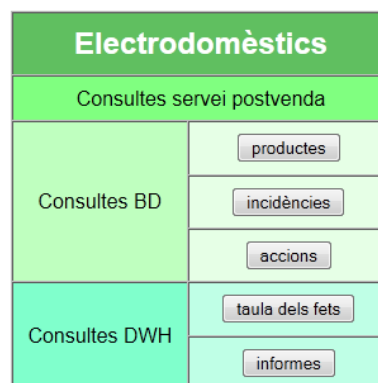
Il·lustració 6.2: Estructura de directoris aplicació Tomcat

Estem seguint els passos explicats en detall al capítol 2, sobre la preparació d'*Apache Tomcat*, però ara adaptant-ho al treball. En l'annex 7, “Pàgines HTML per accedir a la BD i al DWH”, s'inclou la codificació de les pàgines que s'han anomenat:

*MenuInici.html*, *MenuInformes.html* i *EntradaDadesIncid.html*.

### 6.1.1. Menú inicial

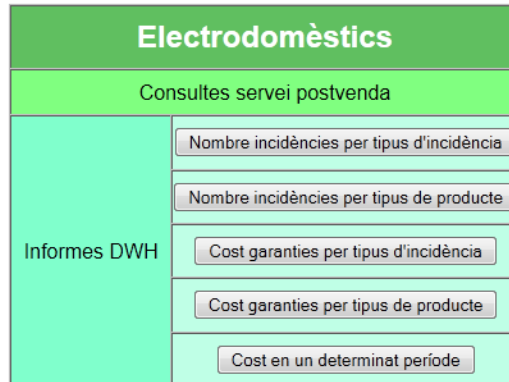
Des de “<http://localhost:8080/UOC-TFG-BBDD/MenuInici.html>” accedim al menú inicial i podem fer les consultes de les principals taules de la BD, com són les taules de productes, incidències i accions, i al DWH per consultar la taula fets i als informes:



Il·lustració 6.3: Web, Menú inicial

### 6.1.2. Menú informes DWH

Des del menú d'informes DWH, que podem accedir des del menú *inicial*, accedirem als diferents informes disponibles:



Il·lustració 6.4: Web, menú informes DWH

En el cas de realitzar un informe en un determinat període, opció "Cost en un determinat període", redirigirem la consulta a una pàgina on ens demana les dates, i tot seguit, permet seleccionar l'informe a obtenir:

**Entrada dates del període a consultar**

Data inici (yyyy-mm-dd):

Data Final (yyyy-mm-dd):

Selecciona l'informe a obtenir:

Il·lustració 6.5: Web, pàgina per especificar període de consulta

## 6.2. Processos del servidor

Utilitzant el Java com a llenguatge de programació que s'executarà al servidor web per centralitzar l'accés al SGBD, preparam dues classes, una per accedir a la BD i l'altra per accedir al DWH, que anomenem: *Electrodomestics.java* i *ElectroDWH.java*, i que compilem i guardem al directori "Tomcat\apache-tomcat-7.0.11\webapps\UOC-TFG-BBDD\WEB-INF\classes\" on tenim el programari *Apache Tomcat*. Com a detall, en l'annex 8, "Detall del codi *ElectroDWH.java*" hi veiem com construïm la pàgina de l'informe "Cost garanties per tipus d'incidència".

### 6.2.1. Configuració del servidor web

Dins del directori "WEB-INF", on hi tenim els directoris "classes" i "lib", hi creem el fitxer "web.xml" amb el contingut necessari per poder accedir als programes que ens donaran accés a la BD i al DWH. Podem veure el codi en l'annex 9 "Fitxer de configuració del servidor web 'web.xml'".

### 6.3. Accés a la BD i al DWH a través del navegador

Un cop ja ho tenim preparat, ja podem procedir a obtenir dades de la BD i del DWH. Primerament engegarem el servidor web (veure funcionament detallat en el capítol 2), i tot seguit accedim al menú inicial, segons hem detallat en aquest mateix capítol a l'apartat "Menú inicial", i a partir d'aquí fàcilment accedim als informes i a les dades.

#### Consulta taula productes

codiProducte	descripcio	tipus	preuCost	preuVenda	duradaGarantia
ZNZ332W00024	NEV-ZRT 332W	0001	220.0	449.0	2
ZNZ332W00025	NEV-ZRT 332W	0001	220.0	459.0	2
E44500X00009	NEV-NEW END44500X	0001	366.50	759.0	2
B3VS3BP00104	PLAT-NET 3VS300BP	0002	160.49	329.0	4
EDL124700056	ROBA-NETA ROMAN-L1247 7KG	0003	197.85	419.0	4
OTV670F00045	ROBA-SECA VOHV670F	0004	147.70	369.0	3
WP050AP01005	GEL-FRED AFG050AP	0005	112.70	269.0	2
TAP220000077	FERRO-ROENT PTSI 2200	0007	8.75	16.70	1
IRRB53100042	ROBOT-NET ROOMBA531	0009	145.75	369.0	3
SMS884500062	ROBOT-ELUX SR8845 Navibot	0009	142.35	329.0	3
LG1929G00199	ONES-ELUX MS 1929G	0010	33.25	65.0	2
TAT4F2V02109	VENT-ADOR Tropica 4F 2V	0011	17.50	34.30	3
MLJ204500007	LIQUS-FRUIT VITAE JU 200045	0014	22.0	44.0	2
SC100MC00012	ROBOT-XEF CF100, MULTICOOKER	0015	74.99	149.0	2
UFCG72200004	OLI-ROENT CG 7220	0016	11.0	21.9	2

#### Nombre incidències per tipus de producte

Tipus producte	Nombre
Cafetera	1
Microones	1
Nevera	3
Rentaplats	1
Ventilador	1

Il·lustració 6.6: Consultes per interfície web

## 7. Valoració econòmica del projecte

A continuació es fa una valoració econòmica aproximada de la implementació del projecte, on es calcula: el cost del maquinari, el cost del programari, i el cost de desenvolupament.

### 7.1. Cost del maquinari

Per poder implementar el projecte a l'empresa, cal un servidor per a la base de dades, i un servidor web. Físicament es podrien instal·lar en una mateixa màquina, però per seguretat cal separar les funcionalitats dels servidors, i comptem que seran 2 màquines noves, que s'instal·laran a l'empresa:

Producte	Màquina	Característiques principals	Preu € (1)
Servidor BD	DELL™ PowerEdge™ T110 (2)	Servidor per a 1 a 15 usuaris. Accés remot per els usuaris. Inclou SO Windows Server® 2008 R2, Foundation Edition English, 64bit.	1.359,00
Servidor web	DELL™ PowerEdge™ T110 (2)	Servidor per a 1 a 15 usuaris. Accés remot per els usuaris. Inclou SO Windows Server® 2008 R2, Foundation Edition English, 64bit.	1.359,00
			<b>2.718,00</b>

Taula 7.1: Cost del maquinari

(1) El preu no inclou l'IVA, ni les despeses de tramesa

(2) [http://configure2.euro.dell.com/dellstore/config.aspx?b=&c=es&cs=esbsdt1&l=es&oc=PE4T110\\_SOL&s=bsd](http://configure2.euro.dell.com/dellstore/config.aspx?b=&c=es&cs=esbsdt1&l=es&oc=PE4T110_SOL&s=bsd)

### 7.2. Cost del programari

Hem seleccionat programari lliure per a implementar el projecte, i el sistema operatiu pels servidors ja va inclòs amb el preu del servidor, així que es preveu un cost zero en programari, que podem detallar en la taula següent:

Producte	Programari	Preu €
Sistema operatiu servidor BD	SO Windows Server® 2008 R2, Foundation Edition English, 64bit.	0,00

Sistema operatiu servidor web	SO Windows Server® 2008 R2, Foundation Edition English, 64bit.	0,00
SGBD	PostgreSQL 9.0.3-1	0,00
Servidor web	Apache Tomcat 7.0.11	0,00
Editor de diagrames ER	RISE Editor 4.3.0.1	0,00
Llenguatge per programar	Java 1.6.0.24	0,00
		<b>0,00</b>

Taula 7.2: Cost del programari

### 7.3. Cost del desenvolupament

A part de desenvolupament i implementació del projecte en sí, també es preveu la instal·lació dels servidors nous, la connexió dels mateixos a la xarxa de l'empresa, i la instal·lació i la posta a punt del programari. En la següent taula donem una aproximació del que pot costar tot el desenvolupament:

Activitat	Hores	Persona	Preu / Hora	Preu €
Gestió del projecte	52	Cap de projecte	58,00	3.016,00
Instal·lació/config. maquinari	16	Tècnic de sistemes	40,00	640,00
Instal·lació/config. programari	16	Tècnic de sistemes	40,00	640,00
Anàlisi	40	Analista	42,00	1.680,00
Disseny	40	Analista	42,00	1.680,00
Desenvolup. / implementació	144	Programador	30,00	4.320,00
Proves	40	Programador	30,00	1.200,00
Documentació	24	Analista	42,00	1.008,00
Formació usuaris	8	Analista	42,00	336,00
Totals	392			<b>14.520,00</b>

Taula 7.3: Cost del desenvolupament

## 8. Conclusions

En aquest treball final de grau (TFG) he desenvolupat una solució, de les moltes possibles, a l'enunciat plantejat per dissenyar i implementar una base de dades relacional per a la gestió d'incidències de productes atenent les especificacions inicialment establertes, i que s'han descrit en la introducció.

He constatat poques variacions en la planificació inicialment prevista en el pla de treball, ja que havia previst totes les tasques necessàries i es van sincronitzar bé amb les fites, sobretot per les marcades pel pla docent, així mateix, vaig tenir molt present la situació personal en la temporització, valorant l'esforç i els costos per poder realitzar totes les activitats previstes i poder construir els productes associats amb el projecte.

Per realitzar el projecte del treball, he seguit el mètode en cascada o cicle de vida clàssic, aplicant en cada tasca les etapes del mètode: definició de requisits, anàlisi, disseny, implementació, proves i manteniment. Però vaig estar estudiant i intentant aplicar el mètode *Scrum*, en les primeres fases de preparació del treball. *Scrum* és un mètode àgil, iteratiu i incremental. El vaig trobar interessant, i vaig creure que podria adaptar-se bé al treball, però la meua falta d'experiència i coneixements en aquest mètode, i sobretot, perquè no era objectiu d'aquest treball aprendre i aplicar un mètode nou, em van fer acabar de decidir per aplicar sense més dilacions el mètode conegut, el mètode en cascada, que és el mètode que per les característiques del treball i el planteig s'hi adaptaven millor, i millor coneixia.

S'han acomplert els objectius bàsics, com eren: analitzar les necessitats bàsiques del sistema a implementar, dissenyar una proposta i implementar-la encapsulant les funcions d'accés a les dades. I com a funcionalitat addicional de valor afegit he desenvolupat un sistema web per a realitzar consultes i obtenir els informes preparats del magatzem de dades, facilitant la interacció de l'usuari amb la BD i el DWH.

També s'han acomplert els objectius específics, i una part important del treball ha estat realitzar-lo amb un SGBD relacional amb el qual no s'hi ha treballat mai abans, com era per a mi el *PostgreSQL*. Si bé té moltes característiques semblants a altres SGBD, com són l'*Oracle* i *Informix*, amb els quals sí que havia treballat, també té unes característiques particulars que he après treballant-hi i documentant-me. He de reconèixer que no ha estat difícil, ja que l'SQL que aplica *PostgreSQL* té poques variants



a l'SQL après amb els altres SGBD, i amb ajuda de la documentació que ofereix el mateix programari i el llibre "*SQL in a Nutshell, Second Edition*" de Kevin E. Kline he pogut solucionar els dubtes.

Finalment, valoro molt positivament el treball, ja que m'ha permès aplicar molts dels coneixements apresos en diverses assignatures, i he pogut constatar que per poder arribar al final del treball m'ha requerit molt de temps i dedicació constant des del primer dia.

## Glossari

<b>ACID</b>	<i>Atomicity, Consistency, Isolation, Durability</i> . Conjunt de propietats que garanteixen les operacions de base de dades es processen de forma fiable.
<b>BD</b>	Base de dades. Conjunt estructurat de dades organitzades segons una estructura coherent, i accessibles des de més d'un programa o aplicació, de manera que qualsevol d'elles pot ésser extreta del conjunt i actualitzada, sense afectar ni l'estructura del conjunt ni les altres dades. En aquest treball considerem les bases de dades del model relacional, que permeten establir interconnexions (relacions) entre les dades (que estan guardades en taules), i treballar amb elles conjuntament.
<b>Base de dades operacional</b>	Base de dades que ajuda en les operacions diàries de negoci d'una organització.
<b>DWH</b>	( <i>Data warehouse</i> ). Magatzem de dades. Base de dades amb informació històrica dissenyada i estructurada per a realitzar-hi consultes de manera eficient.
<b>ER</b>	Model entitat-relació. Model on els objectes o entitats, amb atributs, es vinculen mitjançant relacions.
<b>ETL</b>	( <i>Extract, transform and load</i> ). Processos d'extracció, transformació i càrrega que permeten moure dades des de múltiples fonts, reformatar-les i netejar-les per carregar-les en una base de dades o DWH.
<b>HTML</b>	<i>Hyper Text Markup Language</i> o llenguatge de marcat d'hipertext. Llenguatge de marcat per estructurar textos i relacionar-los en forma d'hipertext molt utilitzat per a la construcció de documents per als web.
<b>JDBC</b>	<i>Java Database Connectivity</i> . És una interfície de programació d'aplicacions (API) que permet a les aplicacions en llenguatge Java accedir mitjançant una interfície comuna a les bases de dades.

<b>JSP</b>	<i>JavaServer Pages</i> . És una tecnologia que permet als desenvolupadors de pàgines web, generar respostes dinàmicament a peticions HTTP.
<b>Magatzem de dades</b>	Veure definició de DWH.
<b>PL/SQL</b>	<i>Procedural Language/Structured Query Language</i> és un llenguatge de programació propietari d'Oracle. PostgreSQL utilitza el PSQL, que és semblant. El llenguatge s'emmagatzema en les bases de dades, i els programes que es realitzen amb aquest llenguatge reben el nom de procediments emmagatzemats.
<b>Servlets</b>	Miniaplicacions de servidor, o <i>servlets</i> en anglès, són objectes Java executats per un servidor d'aplicacions i que responen a invocacions HTTP, servint pàgines web dinàmiques.
<b>SGBD</b>	Sistema de gestió de bases de dades. Conjunt de programes informàtics dissenyats per facilitar la gestió d'un conjunt de dades en una base de dades.
<b>SQL</b>	<i>Structured Query Language</i> o Llenguatge d'interrogació estructurat. És un llenguatge estàndard de comunicació amb bases de dades relacionals.
<b>SQL dinàmic</b>	SQL on es produeix una modificació total o parcial de les sentències durant l'execució del programa.
<b>SQL estàtic</b>	SQL on les sentències usades no canvien durant l'execució del programa.
<b>Scrum</b>	És un mètode àgil, iteratiu i incremental per al desenvolupament de programari.
<b>Sprint backlog</b>	És la llista de tasques que l'equip d' <i>Scrum</i> està realitzant per completar una unitat bàsica de desenvolupament o <i>sprint</i> .
<b>UML</b>	<i>Unified Modeling Language</i> o llenguatge de modelat unificat és un llenguatge per especificar, dissenyar, construir i documentar sistemes, inicialment de programari orientat a objectes.

## Bibliografia

Abelló Gamazo, Alberto. (2003). "*Disseny multidimensional*". Rius Gavídia, Àngels; Serra Vizern, Montse (coordinadores). *Magatzems de dades i models multidimensionals*. Barcelona: Fundació per a la Universitat Oberta de Catalunya

Alex from Question-Defense.com (2010). *Export PostgreSQL Table Data To Excel CSV File From PSQL CLI*. [en línia]. <http://www.question-defense.com/2010/01/08/export-postgresql-table-data-to-excel-csv-file-from-psql-cli> [data de consulta: 26/04/2011].

Apache Software Foundation, The. (2011). *Apache Tomcat*. [en línia]. <http://tomcat.apache.org/> [data de consulta: 25/03/2011].

Costal Costa, Dolors. (2005). "*Disseny de bases de dades*". Sistac Planas, Jaume (coordinador) *Bases de dades I*. Barcelona: Fundació per a la Universitat Oberta de Catalunya

Grubbs, Edwin (2008). *Intro to Stored Procedures in PostgreSQL*. [en línia]. [http://www.day32.com/MySQL/Meetup/Presentations/postgresql\\_stored\\_procedures.pdf](http://www.day32.com/MySQL/Meetup/Presentations/postgresql_stored_procedures.pdf) [data de consulta: 11/04/2011].

Hall, Marty; Krasner, Shlomo. (2008). *Java Programming Resources. A Servlet and JSP Tutorial: Handling Form Data*. [en línia]. <http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-Form-Data.html> [data de consulta: 17/05/2011].

Ippolito, Greg. (2002) *YoLinux Tutorial: Java Servlets, JSP, Jakarta-Tomcat, a Database (PostgreSQL or MySQL), Apache and Linux*. [en línia]. <http://www.yolinux.com/TUTORIALS/LinuxTutorialTomcat.html> [data de consulta 26/03/2011]

Kline, Kevin E.(2004). *SQL in a Nutshell, Second Edition*. Sebastopol: O'Reilly Media, Inc

MediaGroup1 LLC. (2011). *Date Validation using SimpleDateFormat()*. [en línia]. <http://www.dreamincode.net/forums/topic/14886-date-validation-using-simpledateformat/> [data de consulta: 16/05/2011].

plus2net.com. (2011). *Form submitting data to new window and to two different pages*. [en línia]. [http://www.plus2net.com/html\\_tutorial/submit-two.php](http://www.plus2net.com/html_tutorial/submit-two.php) [data de consulta: 18/05/2011].

PostgreSQL (2011). *About*. [en línia]. <http://www.postgresql.org/about/> [data de consulta: 25/03/2011].

PostgreSQL Global Development Group, The. (2010). *PostgreSQL 9.0.3 Documentation*. California: University of California

Rawsthorne, Dan; Shimp, Douglas E. (2009). *Scrum In A Nutshell*. [en línia]. <http://advancedtopicsinscrum.com/wp-content/uploads/2009/04/scrum-in-a-nutshell.pdf> [data de consulta: 11/03/2011].

RISE to Bloome. (2011). *RISE - Simply a better way*. [en línia] [http://www.risetobloome.com/Page\\_1\\_S.aspx?ITEM=1853](http://www.risetobloome.com/Page_1_S.aspx?ITEM=1853) [data de consulta: 03/04/2011]

Saksa, Olli-Pekka. (2009). *Scrum: In Theory & Practice*. [en línia]. <http://www.cs.tut.fi/kurssit/OHJ-3056/luennot/vierailuluennot/2009/VL3-Scrum.pdf> [data de consulta: 27/03/2011].

Wikipedia. (2011). *ACID (atomicity, consistency, isolation, durability)*. [en línia]. <http://en.wikipedia.org/wiki/ACID> [data de consulta: 25/03/2011].



## Annex 2: Codi del programa Java 'MostraBD.java'<sup>4</sup>

```

/* A servlet to display the contents of the PostgreSQL mydatabasename database */
import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class MostraBD extends HttpServlet {
    public String getServletInfo() {
        return "Servlet connects to PostgreSQL database and displays result of a SELECT";
    }

    private Connection dbcon; // Connection

    // "init" sets up a database connection
    public void init(ServletConfig config) throws ServletException {
        String loginUser = "postgres";
        String loginPasswd = "*****";
        String loginUrl = "jdbc:postgresql://localhost:5432/mydatabasename";

        // Load the PostgreSQL driver
        try {
            Class.forName("org.postgresql.Driver");
            dbcon = DriverManager.getConnection(loginUrl, loginUser, loginPasswd);
        } catch (ClassNotFoundException ex) {
            System.err.println("ClassNotFoundException: " + ex.getMessage());
            throw new ServletException("Class not found Error");
        } catch (SQLException ex) {
            System.err.println("SQLException: " + ex.getMessage());
        }
    }

    // Use http GET
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html"); // Response mime type

        // Output stream to STDOUT
        PrintWriter out = response.getWriter();

        out.println("<HTML><Head><Title>mydatabasename</Title></Head>");
        out.println("<Body><H1>mydatabasename</H1>");

        try {
            // Declare our statement
            Statement statement = dbcon.createStatement();

            String query = "SELECT name, dept, ";
            query += "      jobtitle ";
            query += "FROM   employee ";

            // Perform the query

```

<sup>4</sup> Adaptació de l'exemple extret del tutorial de Greg Ippolito a <http://www.yolinux.com/>

```
ResultSet rs = statement.executeQuery(query);

out.println("<table border>");

// Iterate through each row of rs
while (rs.next()) {
    String m_name = rs.getString("name");
    String m_dept = rs.getString("dept");
    String m_jobtitle = rs.getString("jobtitle");
    out.println("<tr>" +
        "<td>" + m_name + "</td>" +
        "<td>" + m_dept + "</td>" +
        "<td>" + m_jobtitle + "</td>" +
        "</tr>");
}

out.println("</table></body></html>");
statement.close();
} catch (Exception ex) {
    out.println("<HTML>" +
        "<Head><Title>" +
        "mydatabasename: Error" +
        "</Title></Head>\n<Body>" +
        "<P>SQL error in doGet: " +
        ex.getMessage() + "</P></Body></HTML>");
    return;
}
out.close();
}
```



## Annex 3: Codi SQL per la creació de les taules de la BD

```

--CREATE DATABASE electrodomestics

-----
-- TAULA gamma -- Gammes d'electrodomestics que es fabricaran
-----
CREATE TABLE gamma (
  codiGamma CHAR(1) CONSTRAINT PK_gamma PRIMARY KEY,
  descripcio VARCHAR(16) CONSTRAINT NN_gamma_dsc NOT NULL,
  CONSTRAINT UN_gamma_dsc UNIQUE (descripcio)
);
-----
-- TAULA tipus_prd -- Tipus de productes que es fabricaran
-----
CREATE TABLE tipus_prd (
  codiTipusPrd CHAR(4) CONSTRAINT PK_tipus_prd PRIMARY KEY,
  gamma CHAR(1) CONSTRAINT FK_tipus_prd_gamma REFERENCES gamma(codiGamma)
  CONSTRAINT NN_tipus_prd_gamma NOT NULL,
  descripcio VARCHAR(32) CONSTRAINT NN_tipus_prd_dsc NOT NULL,
  CONSTRAINT UN_tipus_prd_dsc UNIQUE (descripcio)
);
-----
-- TAULA producte -- Productes
-----
CREATE TABLE producte (
  codiProducte CHAR(12) CONSTRAINT PK_producte PRIMARY KEY,
  descripcio VARCHAR(32) CONSTRAINT NN_producte_dsc NOT NULL,
  tipus CHAR(4) CONSTRAINT FK_producte_tip REFERENCES tipus_prd(codiTipusPrd)
  CONSTRAINT NN_producte_tip NOT NULL,
  preuCost NUMERIC CONSTRAINT NN_producte_prc NOT NULL
  CHECK (preuCost > 0),
  preuVenda NUMERIC CONSTRAINT NN_producte_prv NOT NULL
  CONSTRAINT CH_producte_prv CHECK (preuVenda > 0),
  duradaGarantia INTEGER CONSTRAINT NN_producte_dGr NOT NULL
);
-----
-- TAULA client -- Clients, empreses que ens compren els productes
-----
CREATE TABLE client (
  codiClient CHAR(8) CONSTRAINT PK_client PRIMARY KEY,
  nif CHAR(9) CONSTRAINT NN_client_nif NOT NULL,
  nom VARCHAR(42) CONSTRAINT NN_client_nom NOT NULL,
  adress VARCHAR(60) CONSTRAINT NN_client_adr NOT NULL,
  telefon CHAR(9) CONSTRAINT NN_client_tel NOT NULL,
  descompte NUMERIC DEFAULT 0.0
  CONSTRAINT NN_client_des NOT NULL,
  CONSTRAINT CH_client_des CHECK (descompte >= 0),
  CONSTRAINT UN_client_nif UNIQUE (nif)
);
-----
-- TAULA empleat -- Empleats de l'empresa
-----
CREATE TABLE empleat (
  codiEmpleat CHAR(4) CONSTRAINT PK_empleat PRIMARY KEY,
  nif CHAR(9) CONSTRAINT NN_empleat_nif NOT NULL,
  nom VARCHAR(42) CONSTRAINT NN_empleat_nom NOT NULL,

```

```

adress    VARCHAR(60) CONSTRAINT NN_empleat_adr NOT NULL,
telefon  CHAR(9),
nss      CHAR(14)   CONSTRAINT NN_empleat_nss NOT NULL,
                                CONSTRAINT UN_empleat_nss UNIQUE (nss),
                                CONSTRAINT UN_empleat_nif UNIQUE (nif)
);
-----
-- TAULA venedor -- Venedors, subtipus d'empleats
-----
CREATE TABLE venedor (
  codiEmpleat CHAR(4) CONSTRAINT PK_venedor PRIMARY KEY
                                CONSTRAINT FK_venedor_cod
                                REFERENCES empleat(codiEmpleat),
  comissioVendes NUMERIC CONSTRAINT NN_venedor_com NOT NULL
);
-----
-- TAULA agent_telefonic -- Agents telefonics, subtipus d'empleats
-----
CREATE TABLE agent_telefonic (
  codiEmpleat CHAR(4) CONSTRAINT PK_agent_telefonic PRIMARY KEY
                                CONSTRAINT FK_agent_telefonic_cod
                                REFERENCES empleat(codiEmpleat),
  telefonDepartament CHAR(9) CONSTRAINT NN_agent_telefonic_tlf NOT NULL
);
-----
-- TAULA enginyer -- Enginyers, subtipus d'empleats
-----
CREATE TABLE enginyer (
  codiEmpleat CHAR(4)   CONSTRAINT PK_enginyer PRIMARY KEY
                                CONSTRAINT FK_enginyer_cod REFERENCES empleat(codiEmpleat),
  especialitat VARCHAR(60) CONSTRAINT NN_enginyer_tlf NOT NULL
);
-----
-- TAULA especialitzacio -- Relacio entre enginyer i gammes en que estan especialitzats
-----
CREATE TABLE especialitzacio (
  enginyer CHAR(4) CONSTRAINT FK_especialitzacio_eng REFERENCES enginyer(codiEmpleat)
                                CONSTRAINT NN_especialitzacio_eng NOT NULL,
  gamma CHAR(1) CONSTRAINT FK_especialitzacio_gam REFERENCES gamma(codiGamma)
                                CONSTRAINT NN_especialitzacio_gam NOT NULL,
                                CONSTRAINT PK_especialitzacio PRIMARY KEY(enginyer, gamma)
);
-----
-- TAULA factura -- Factures emeses per l'empresa que certifiquen una venda
-----
CREATE TABLE factura (
  codiFactura CHAR(10) CONSTRAINT PK_factura_num PRIMARY KEY,
  venedor CHAR(4) CONSTRAINT FK_factura_ven REFERENCES venedor(codiEmpleat)
                                CONSTRAINT NN_factura_ven NOT NULL,
  data DATE CONSTRAINT NN_factura_dat NOT NULL
);
-----
-- TAULA adquisicio -- Relacio entre factura, client i producte
-----
CREATE TABLE adquisicio (
  producte CHAR(12) CONSTRAINT FK_adquisicio_prd REFERENCES producte(codiProducte)
                                CONSTRAINT NN_adquisicio_prd NOT NULL,
  factura CHAR(10) CONSTRAINT FK_adquisicio_fct REFERENCES factura(codiFactura)
                                CONSTRAINT NN_adquisicio_fct NOT NULL,
  client CHAR(8) CONSTRAINT FK_adquisicio_clt REFERENCES client(codiClient)
                                CONSTRAINT NN_adquisicio_clt NOT NULL,
                                CONSTRAINT PK_adquisicio PRIMARY KEY(producte, factura, client),
);

```

```

                CONSTRAINT UN_adquisicio    UNIQUE (producte)
);
-----
-- TAULA tipus_inc -- Codificacio dels tipus d'incidencies
-----
CREATE TABLE tipus_inc (
  codiTipusInc      CHAR(4)    CONSTRAINT PK_tipus_inc    PRIMARY KEY,
  descripcio        VARCHAR(32) CONSTRAINT NN_tipus_inc_dsc NOT NULL,
  garantiaAplicable CHAR(1)    CONSTRAINT NN_tipus_inc_gar NOT NULL,
                                     CONSTRAINT CH_tipus_inc_gar
                                     CHECK(garantiaAplicable IN ('S', 'N'))
);
-----
-- TAULA incidencia -- Incidencies postvenda
-----
CREATE TABLE incidencia (
  codiIncidencia   CHAR(12)   CONSTRAINT PK_incidencia    PRIMARY KEY,
  descripcio        VARCHAR(80) CONSTRAINT NN_incidencia_dsc NOT NULL,
  producte          CHAR(12)   CONSTRAINT FK_incidencia_prd REFERENCES producte(codiProducte)
                                     CONSTRAINT NN_incidencia_prd NOT NULL,
  factura           CHAR(10)   CONSTRAINT FK_incidencia_fct REFERENCES factura(codiFactura)
                                     CONSTRAINT NN_incidencia_fct NOT NULL,
  tipusIncidencia  CHAR(4)    CONSTRAINT FK_incidencia_tip REFERENCES tipus_inc(codiTipusInc)
                                     CONSTRAINT NN_incidencia_tip NOT NULL,
  dataObertura     DATE       CONSTRAINT NN_incidencia_dt0 NOT NULL,
  dataTancament    DATE,
                                     CONSTRAINT CH_incidencia_dtT
                                     CHECK (dataTancament >= dataObertura)
);
-----
-- TAULA comunicacio -- Relacio de comunicacio d'una incidencia
-----
CREATE TABLE comunicacio (
  incidencia        CHAR(12)   CONSTRAINT FK_comunicacio_inc
                                     REFERENCES incidencia(codiIncidencia)
                                     CONSTRAINT NN_comunicacio_inc NOT NULL,
  client            CHAR(8)    CONSTRAINT FK_comunicacio_clt
                                     REFERENCES client(codiClient)
                                     CONSTRAINT NN_comunicacio_clt NOT NULL,
  agentTelefonic   CHAR(4)    CONSTRAINT FK_comunicacio_atl
                                     REFERENCES agent_telefonic(codiEmpleat)
                                     CONSTRAINT NN_comunicacio_atl NOT NULL,
                                     CONSTRAINT PK_comunicacio
                                     PRIMARY KEY(incidencia, client, agentTelefonic)
);
-----
-- TAULA accio -- Accions que es realitzaran en una incidencia
-----
CREATE TABLE accio (
  codiAccio        CHAR(12)   CONSTRAINT PK_accio        PRIMARY KEY,
  incidencia        CHAR(12)   CONSTRAINT FK_accio_inc    REFERENCES incidencia(codiIncidencia)
                                     CONSTRAINT NN_accio_inc    NOT NULL,
  enginyer         CHAR(4)    CONSTRAINT FK_accio_eng    REFERENCES enginyer(codiEmpleat)
                                     CONSTRAINT NN_accio_eng    NOT NULL,
  descripcio        VARCHAR(80) CONSTRAINT NN_accio_dsc  NOT NULL,
  dataInici        DATE       CONSTRAINT NN_accio_dtI    NOT NULL,
  dataFi           DATE       CONSTRAINT CH_accio_dtF    CHECK (dataFi >= dataInici),
  cost             NUMERIC     DEFAULT 0.0
                                     CONSTRAINT CH_accio_cst    CHECK (cost >= 0)
);

```

## Annex 4: Procediment emmagatzemat de la taula 'accio'

```

-----
-- Procediments TAULA accio
-----
CREATE OR REPLACE FUNCTION accio_stamp() RETURNS trigger AS $accio_stamp$
DECLARE
    errorFound BOOLEAN;
    errorText TEXT;
    logCall TEXT;
    logText TEXT;
    logText2 TEXT;
    regFound TEXT;
    row RECORD;
BEGIN
    IF (TG_OP = 'DELETE') THEN
        ROW = OLD;
    ELSE -- 'INSERT' & 'UPDATE'
        ROW = NEW;
        -- Comprovar que els valors son correctes
        errorFound = FALSE;
        IF NEW.codiaccio IS NULL THEN errorFound = TRUE; errorText='codiaccio';
        ELSIF NEW.incidencia IS NULL THEN errorFound = TRUE; errorText='incidencia';
        ELSIF NEW.engineyer IS NULL THEN errorFound = TRUE; errorText='engineyer';
        ELSIF NEW.dataInici IS NULL THEN errorFound = TRUE; errorText='dataInici';
        END IF;
        IF errorFound THEN
            RAISE NOTICE '-- ELS VALORS: codiaccio, incidencia, engineyer i dataInici';
            RAISE NOTICE '-- SON OBLIGATORIS.';
            RAISE NOTICE '-- OPERACIO : % a %', TG_OP, TG_TABLE_NAME;
            RAISE NOTICE '-- REGISTRE : %', NEW;
            RAISE EXCEPTION '-- Error trobat valor NUL a %', errorText;
        END IF;

        SELECT INTO regFound codiIncidencia FROM incidencia WHERE codiIncidencia = NEW.incidencia;
        IF NOT FOUND THEN
            RAISE EXCEPTION
                '-- Error gravant accio: incidencia % no valida, no es a taula incidencia',
                NEW.incidencia;
        END IF;

        SELECT INTO regFound codiEmpleat FROM engineyer WHERE codiEmpleat = NEW.engineyer;
        IF NOT FOUND THEN
            RAISE EXCEPTION
                '-- Error gravant accio: engineyer % no valid, no es a taula engineyer',
                NEW.engineyer;
        END IF;

        IF NEW.dataFi IS NOT NULL THEN
            IF NEW.dataFi < NEW.dataInici THEN
                RAISE EXCEPTION '-- Error gravant accio: dataFi (%) < dataInici (%)',
                    NEW.dataFi, NEW.dataInici;
            END IF;
        END IF;

        IF NEW.cost < 0 THEN
            RAISE EXCEPTION '-- Error gravant accio: cost accio (%) < 0', NEW.cost;
        END IF;
    END IF;
END IF;

```

```

SELECT INTO regFound dataObertura FROM incidencia WHERE codiIncidencia = NEW.incidencia;
IF NEW.dataInici < to_date(regFound,'YYYY-MM-DD') THEN
  RAISE EXCEPTION
  '-- Error gravant accio: dataInici accio (%) < data inici incidencia (%)',
  NEW.dataInici, regFound;
END IF;

SELECT INTO regFound dataTancament FROM incidencia WHERE codiIncidencia = NEW.incidencia;
IF regFound IS NOT NULL THEN
  IF NEW.dataInici > to_date(regFound,'YYYY-MM-DD') THEN
    RAISE EXCEPTION
    '-- Error gravant accio: dataInici accio (%) > data tancament incidencia (%)',
    NEW.dataInici, regFound;
  ELSIF NEW.dataFi > to_date(regFound,'YYYY-MM-DD') THEN
    RAISE EXCEPTION
    '-- Error gravant accio: dataFi accio (%) > data tancament incidencia (%)',
    NEW.dataFi, regFound;
  END IF;
END IF;
END IF;

logText = ROW.codiAccio ||';'|| ROW.incidencia ||';'|| ROW.enginyer ||';'
          || ROW.dataInici ||';';
IF ROW.dataFi IS NOT NULL THEN logText = logText || ROW.dataFi ||';';
ELSE logText = logText ||';';
END IF;
logText = logText || ROW.cost ||';';

IF (TG_OP = 'UPDATE') THEN
  logText2 = OLD.codiAccio ||';'|| OLD.incidencia ||';'|| OLD.enginyer ||';'
            || OLD.dataInici ||';';
  IF OLD.dataFi IS NOT NULL THEN logText2 = logText2 || OLD.dataFi ||';';
  ELSE logText2 = logText2 ||';';
  END IF;
  logText = logText2 ||'-->'; logText;
END IF;
logCall = gravar_log(TG_OP, TG_TABLE_NAME, logText);
RETURN ROW;
END;
$accio_stamp$ LANGUAGE plpgsql;
-----
DROP TRIGGER IF EXISTS accio_stamp ON accio;
CREATE TRIGGER accio_stamp BEFORE INSERT OR UPDATE OR DELETE ON accio
FOR EACH ROW EXECUTE PROCEDURE accio_stamp();

```

## Annex 5: Codi SQL per la creació de les taules del DWH

```

--CREATE DATABASE electroDWH

-----
-- Neteja de taules
-----
DROP TABLE IF EXISTS DWH_res;
DROP TABLE IF EXISTS DWH_acc;
DROP TABLE IF EXISTS DWH_inc;
DROP TABLE IF EXISTS DWH_prd;

-----
-- TAULA DWH_prd -- producte DWH --
-----
CREATE TABLE DWH_prd (
  DWH_prd_codi CHAR(12) CONSTRAINT PK_DWH_prd PRIMARY KEY,
  DWH_prd_descr VARCHAR(32) CONSTRAINT NN_DWH_prd_descr NOT NULL,
  DWH_prd_tipus VARCHAR(32) CONSTRAINT NN_DWH_prd_tipus NOT NULL,
  DWH_prd_gamma VARCHAR(16) CONSTRAINT NN_DWH_prd_gamma NOT NULL,
  DWH_prd_data DATE CONSTRAINT NN_DWH_prd_data NOT NULL,
  DWH_prd_tmGar INTEGER CONSTRAINT NN_DWH_prd_tmGar NOT NULL
);

-----
-- TAULA DWH_inc -- incidència DWH --
-----
CREATE TABLE DWH_inc (
  DWH_inc_codi CHAR(12) CONSTRAINT PK_DWH_inc PRIMARY KEY,
  DWH_inc_prod CHAR(12) CONSTRAINT FK_DWH_inc_prod REFERENCES DWH_prd(DWH_prd_codi)
  CONSTRAINT NN_DWH_inc_prod NOT NULL,
  DWH_inc_tipus VARCHAR(32) CONSTRAINT NN_DWH_inc_tipus NOT NULL,
  DWH_inc_dtIni DATE CONSTRAINT NN_DWH_inc_dtIni NOT NULL,
  DWH_inc_dtFi DATE,
  DWH_inc_InGar CHAR(1) CONSTRAINT NN_DWH_inc_InGar NOT NULL,
  CONSTRAINT CH_DWH_inc_InGar
  CHECK(DWH_inc_InGar IN ('S', 'N'))
);

-----
-- TAULA DWH_acc -- acció DWH --
-----
CREATE TABLE DWH_acc (
  DWH_acc_codi CHAR(12) CONSTRAINT PK_DWH_acc PRIMARY KEY,
  DWH_acc_incid CHAR(12) CONSTRAINT FK_DWH_acc_incid REFERENCES DWH_inc(DWH_inc_codi)
  CONSTRAINT NN_DWH_acc_incid NOT NULL,
  DWH_acc_cost NUMERIC CONSTRAINT NN_DWH_acc_cost NOT NULL
);

-----
-- TAULA DWH_res -- resultats DWH -- Fet --
-----
CREATE TABLE DWH_res (
  DWH_res_prod CHAR(12) CONSTRAINT FK_DWH_res_prod REFERENCES DWH_prd(DWH_prd_codi)
  CONSTRAINT NN_DWH_res_prod NOT NULL,
  DWH_res_incid CHAR(12) CONSTRAINT FK_DWH_res_inci REFERENCES DWH_inc(DWH_inc_codi)
  CONSTRAINT NN_DWH_res_inci NOT NULL,
  DWH_res_accio CHAR(12) CONSTRAINT FK_DWH_res_accio REFERENCES DWH_acc(DWH_acc_codi)
  CONSTRAINT NN_DWH_res_accio NOT NULL,
  CONSTRAINT PK_DWH_res PRIMARY KEY(DWH_res_prod,
  DWH_res_incid, DWH_res_accio)
);

```

## Annex 6: Procediments emmagatzemats de la taula 'DWH\_acc'

```

-----
-- Procediments TAULA DWH_acc
-----
-- Quan s'actualitzi la taula,
-- es mantindra actualitzada la taula de resultats DWH_res
-----
CREATE OR REPLACE FUNCTION DWH_acc_stamp() RETURNS trigger AS $DWH_acc_stamp$
DECLARE
  errorFound BOOLEAN;
  errorText TEXT;
  logCall TEXT;
  logText TEXT;
  regFound TEXT;
BEGIN
  -- Comprovem que la incidència existeix
  SELECT INTO regFound DWH_inc_codi FROM DWH_inc WHERE DWH_inc_codi = NEW.DWH_acc_incid;
  IF NOT FOUND THEN
    RAISE EXCEPTION
      '-- Error gravant DWH_acc: incidència % no vàlida, no a taula DWH_inc',
      NEW.DWH_acc_incid;
  END IF;

  IF (TG_OP = 'UPDATE') THEN
    -- Actualitzem el registre a la taula de resultats
    UPDATE DWH_res
      SET DWH_res_prod = (SELECT DWH_inc_prod
                          FROM DWH_inc
                          WHERE DWH_inc_codi = NEW.DWH_acc_incid),
          DWH_res_incid = NEW.DWH_acc_incid,
          DWH_res_accio = NEW.DWH_acc_codi
      WHERE DWH_res_prod = (SELECT DWH_inc_prod FROM DWH_inc
                            WHERE DWH_inc_codi = OLD.DWH_acc_incid)
          AND DWH_res_incid = OLD.DWH_acc_incid
          AND DWH_res_accio = OLD.DWH_acc_codi;
  ELSE -- (TG_OP = 'INSERT') THEN
    -- Gravem el registre a la taula de resultats
    INSERT INTO DWH_res (DWH_res_prod, DWH_res_incid, DWH_res_accio)
      VALUES ((SELECT DWH_inc_prod
                FROM DWH_inc
                WHERE DWH_inc_codi = NEW.DWH_acc_incid),
              NEW.DWH_acc_incid, NEW.DWH_acc_codi);
  END IF;

  RETURN NEW;
END;
$DWH_acc_stamp$ LANGUAGE plpgsql;
-----
DROP TRIGGER IF EXISTS DWH_acc_stamp ON DWH_acc;
CREATE TRIGGER DWH_acc_stamp AFTER INSERT OR UPDATE ON DWH_acc
  FOR EACH ROW EXECUTE PROCEDURE DWH_acc_stamp();
-----
CREATE OR REPLACE FUNCTION DWH_acc_delete() RETURNS trigger AS $DWH_acc_delete$
DECLARE
  errorFound BOOLEAN;
  errorText TEXT;
  logCall TEXT;
  logText TEXT;

```

```
regFound TEXT;
BEGIN
-- Comprovem que la incidència existeix
SELECT INTO regFound DWH_inc_codi FROM DWH_inc WHERE DWH_inc_codi = OLD.DWH_acc_incid;
IF NOT FOUND THEN
  RAISE EXCEPTION
    '-- Error gravant DWH_acc: incidència % no vàlida, no a taula DWH_inc',
    OLD.DWH_acc_incid;
END IF;

-- Esborrem el registre a la taula de resultats
DELETE FROM DWH_res
  WHERE DWH_res_prod = (SELECT DWH_inc_prod FROM DWH_inc
                        WHERE DWH_inc_codi = OLD.DWH_acc_incid)
    AND DWH_res_incid = OLD.DWH_acc_incid
    AND DWH_res_accio = OLD.DWH_acc_codi;

RETURN OLD;
END;
$DWH_acc_delete$ LANGUAGE plpgsql;
-----
DROP TRIGGER IF EXISTS DWH_acc_delete ON DWH_acc;
CREATE TRIGGER DWH_acc_delete BEFORE DELETE ON DWH_acc
  FOR EACH ROW EXECUTE PROCEDURE DWH_acc_delete();
=====
```



## Annex 7: Pàgines HTML per accedir a la BD i al DWH

### • MenuInici.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Menú inicial</TITLE>
</HEAD>
<BODY><FORM>
  <FONT FACE="ARIAL"><BASEFONT SIZE="4">
  <TABLE BORDER="1" CELLPADDING="0" CELLSPACING="0" WIDTH="300" HEIGHT="300">
    <TR >
      <TD ALIGN="CENTER" COLSPAN="2" WIDTH="300" BGCOLOR="#60BF60">
        <FONT FACE="ARIAL" SIZE="5" COLOR="WHITE"><B>Electrodomèstics</B></FONT>
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" COLSPAN="2" WIDTH="300" BGCOLOR="#80FF80">
        Consultes servei postvenda</TD>
    </TR><TR>
      <TD ALIGN="CENTER" ROWSPAN="3" WIDTH="140" BGCOLOR="#BFFF6F">Consultes BD</TD>
      <TD ALIGN="CENTER" WIDTH="160" BGCOLOR="#E6FFE6">
        <INPUT type="button" value="productes"
          onClick="window.location.href='./Electrodomestics?consulta=productes'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" BGCOLOR="#E6FFE6">
        <INPUT type="button" value="incidències"
          onClick="window.location.href='./Electrodomestics?consulta=incidències'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" BGCOLOR="#E6FFE6">
        <INPUT type="button" value="accions"
          onClick="window.location.href='./Electrodomestics?consulta=accions'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" ROWSPAN="2" WIDTH="140" BGCOLOR="#80FFCC">Consultes DWH</TD>
      <TD ALIGN="CENTER" WIDTH="160" BGCOLOR="#BFFF6F">
        <INPUT type="button" value="taula dels fets"
          onClick="window.location.href='./ElectroDWH?consulta=fets'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" BGCOLOR="#BFFF6F">
        <INPUT type="button" value="informes"
          onClick="window.location.href='./MenuInformes.html'">
      </TD>
    </TR>
  </TABLE>
</FONT>
</FORM></BODY></HTML>

```

• **MenuInformes.html**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Menú informes</TITLE>
</HEAD>

<BODY><FORM>
  <FONT FACE="ARIAL"><BASEFONT SIZE="4">
  <TABLE BORDER="1" CELLPADDING="0" CELLSPACING="0" WIDTH="400" HEIGHT="300">
    <TR >
      <TD ALIGN="CENTER" COLSPAN="2" WIDTH="400" BGCOLOR="#60BF60">
        <FONT FACE="ARIAL" SIZE="5" COLOR="WHITE"><B>Electrodomèstics</B></FONT>
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" COLSPAN="2" WIDTH="400" BGCOLOR="#80FF80">
        Consultes servei postvenda</TD>
    </TR><TR>
      <TD ALIGN="CENTER" ROWSPAN="5" WIDTH="140" BGCOLOR="#80FFCC">Informes DWH</TD>
      <TD ALIGN="CENTER" WIDTH="260" BGCOLOR="#BFFFE6">
        <INPUT type="button" value="Nombre incidències per tipus d'incidència"
          onClick="window.location.href='./ElectroDWH?consulta=informe_1'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" BGCOLOR="#BFFFE6">
        <INPUT type="button" value="Nombre incidències per tipus de producte"
          onClick="window.location.href='./ElectroDWH?consulta=informe_2'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" BGCOLOR="#BFFFE6">
        <INPUT type="button" value="Cost garanties per tipus d'incidència"
          onClick="window.location.href='./ElectroDWH?consulta=informe_3'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" BGCOLOR="#BFFFE6">
        <INPUT type="button" value="Cost garanties per tipus de producte"
          onClick="window.location.href='./ElectroDWH?consulta=informe_4'">
      </TD>
    </TR><TR>
      <TD ALIGN="CENTER" BGCOLOR="#BFFFE6">
        <INPUT type="button" value="Cost en un determinat període"
          onClick="window.location.href='./EntradaDatesIncid.html'">
      </TD>
    </TR>
  </TABLE>
  <BR>
  <TABLE BORDER=0>
    <TR>
      <TD><INPUT TYPE="button" VALUE="Inici"
        onClick="window.location.href='./MenuInici.html'"></TD>
      <TD></TD>
      <TD></TD>
    </TR>
  </TABLE>
</FONT>
</FORM></BODY></HTML>

```

## • EntradaDatesIncid.html

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Entrada dates del període a consultar</TITLE>
</HEAD>

<BODY BGCOLOR="#FDF5E6">
<H2>Entrada dates del període a consultar</H2>

<FORM NAME=f1
  ACTION="/ElectroDWH?consulta=informe_5"
  METHOD="POST">
  <FONT FACE="Andale Mono, Courier New"><BASEFONT SIZE="4">

  <BR>
  Data inici (yyyy-mm-dd):
  <INPUT TYPE="TEXT" NAME="FormDataInici" VALUE="2010-01-01"><BR>

  <BR>
  Data Final (yyyy-mm-dd):
  <INPUT TYPE="TEXT" NAME="FormDataFinal" VALUE="2011-12-31"><BR>

  <BR><BR>
  Selecciona l'informe a obtenir:<BR><BR>
  <input type='submit' value='Cost de les incidències'
    onclick="f1.action='/ElectroDWH?consulta=informe_5';return true;">
  <input type='submit' value='Cost de les garanties'
    onclick="f1.action='/ElectroDWH?consulta=informe_6'; return true;">
  <BR><BR>
  <INPUT TYPE="BUTTON" VALUE="Enrera" onClick="history.go(-1);return true;">
  </FONT>
</FORM>

</BODY>
</HTML>

```

**Annex 8: Detall del codi del programa Java 'ElectroDWH.java'**

```

/* A servlet to display the contents of the PostgreSQL database */

import java.io.*;
import java.sql.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ElectroDWH extends HttpServlet
{
    public String getServletInfo() {
        return "Servlet connects to PostgreSQL database and displays result of a SELECT";
    }

    private Connection dbcon; // Connection

    // "init" sets up a database connection
    public void init(ServletConfig config) throws ServletException {
        String loginUser = "postgres";
        String loginPasswd = "SQL";
        String loginUrl = "jdbc:postgresql://localhost:5432/electroDWH";

        // Load the PostgreSQL driver
        try {
            Class.forName("org.postgresql.Driver");
            dbcon = DriverManager.getConnection(loginUrl, loginUser, loginPasswd);
        } catch (ClassNotFoundException ex) {
            System.err.println("ClassNotFoundException: " + ex.getMessage());
            throw new ServletException("Class not found Error");
        } catch (SQLException ex) {
            System.err.println("SQLException: " + ex.getMessage());
        }
    }

    // Use http GET
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException, ServletException {
        response.setContentType("text/html"); // Response mime type

        // Output stream to STDOUT
        PrintWriter out = response.getWriter();

        out.println("<HTML><Head><Title>ElectroDWH</Title></Head>");

        try {
            String CONSULTA = request.getParameter("consulta");
            if (CONSULTA != null) {

                // Declare our statement
                Statement statement = dbcon.createStatement();
                out.println("<table border>");

                if (CONSULTA.equals("fets")) {
                    // ...
                } else if (CONSULTA.equals("informe_1")) {
                    // ...
                }
            }
        }
    }
}

```

```

    } else if (CONSULTA.equals("informe_2")) {
        // ...
    } else if (CONSULTA.equals("informe_3")) {
        out.println("<Body><H2>Cost garanties per tipus d'incidència</H2>");

        String query = "SELECT DWH_inc_tipus AS Tipus_incidencia,";
        query += "      sum(DWH_acc_cost) as Cost_garantia";
        query += " FROM DWH_res, DWH_inc, DWH_acc, DWH_prd";
        query += " WHERE DWH_res_incid = DWH_inc_codi";
        query += "       AND DWH_res_accio = DWH_acc_codi";
        query += "       AND DWH_res_prod = DWH_prd_codi";
        query += "       AND DWH_inc_ingar = 'S'";
        query += "       AND DWH_inc_dtIni <= (DWH_prd_data + (DWH_prd_tmgar *
interval '1 year'))";
        query += " GROUP BY DWH_inc_tipus";

        out.println("<tr BGCOLOR=#80FFCC>" +
            "<td>Tipus incidència</td>" +
            "<td>Cost garantia</td>" +
            "</tr>");

        // Perform the query
        ResultSet rs = statement.executeQuery(query);

        // Iterate through each row of rs
        while (rs.next()) {
            String res_tip_inc = rs.getString("Tipus_incidencia");
            String res_tip_cst = rs.getString("Cost_garantia");
            out.println("<tr BGCOLOR=#BFFF66>" +
                "<td>" + res_tip_inc + "</td>" +
                "<td align=right>" + res_tip_cst + "</td>" +
                "</tr>");
        }
    } else if (CONSULTA.equals("informe_4")) {
        // ...
    } else if (CONSULTA.equals("informe_5")) {
        // ...
    } else if (CONSULTA.equals("informe_6")) {
        // ...
    } else {
        out.println("Consulta '" + CONSULTA + "' no prevista");
    }

    out.println("</table><br><table border=0><tr>");
    out.println("<td><INPUT TYPE=button VALUE=Inici"
onClick="window.location.href='./MenuInici.html'"></td>");
    out.println("<td></td>");
    out.println("<td><INPUT TYPE=button VALUE=Enrera"
onClick="history.go(-1);return true;"></td>");
    out.println("</tr></table></body></html>");
    statement.close();
} else {
    out.println("No s'ha indicat la consulta a fer. Utilitza el format URL?
consulta=xxx");
}

} catch (Exception ex) {
    out.println("<HTML>" +
        "<Head><Title>" +
        "*** Error ***" +
        "</Title></Head>\n<Body>" +
        "<P>SQL error in doGet: " +

```

```

        ex.getMessage() +
        "</P></Body></HTML>");
    }
    return;
}
out.close();
}

public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    doGet(request, response);
}

// Adaptació: http://www.dreamincode.net/forums/topic/14886-date-validation-using-
simpledateformat/
// date validation using SimpleDateFormat
// it will take a string and make sure it's in the proper
// format as defined by you, and it will also make sure that
// it's a legal date
public boolean isValidDate(String date) {
    // set date format, this can be changed to whatever format
    // you want, MM-dd-yyyy, MM.dd.yyyy, dd.MM.yyyy etc.
    // you can read more about it here:
    // http://java.sun.com/j2se/1.4.2/docs/api/index.html

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    // declare and initialize testDate variable, this is what will hold
    // our converted string

    java.util.Date testDate = null;

    // we will now try to parse the string into date form
    try {
        testDate = sdf.parse(date);
    }

    // if the format of the string provided doesn't match the format we
    // declared in SimpleDateFormat() we will get an exception
    catch (ParseException e) {
        // errorMessage = "the date you provided is in an invalid date format.";
        return false;
    }

    // dateFormat.parse will accept any date as long as it's in the format
    // you defined, it simply rolls dates over, for example, december 32
    // becomes jan 1 and december 0 becomes november 30
    // This statement will make sure that once the string
    // has been checked for proper formatting that the date is still the
    // date that was entered, if it's not, we assume that the date is invalid
    if (!sdf.format(testDate).equals(date)) {
        // errorMessage = "The date that you provided is invalid.";
        return false;
    }

    // if we make it to here without getting an error it is assumed that
    // the date was a valid one and that it's in the proper format
    return true;
} // end isValidDate
}

```

## Annex 9: Fitxer de configuració del servidor web 'web.xml'

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  version="3.0"
  metadata-complete="true">

  <description>
    Servlet and JSP - TFG - Antoni Garriga Rovira
  </description>
  <display-name>Servlet and JSP - TFG - Antoni Garriga Rovira</display-name>

  <servlet>
    <servlet-name>Electrodomestics</servlet-name>
    <servlet-class>Electrodomestics</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Electrodomestics</servlet-name>
    <url-pattern>/Electrodomestics</url-pattern>
  </servlet-mapping>

  <servlet>
    <servlet-name>ElectroDWH</servlet-name>
    <servlet-class>ElectroDWH</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ElectroDWH</servlet-name>
    <url-pattern>/ElectroDWH</url-pattern>
  </servlet-mapping>

</web-app>
```

## Annex 10: Relació de fitxers adjunts a la memòria del treball

Aquest document, anomenat memòria de treball, va acompanyat d'un conjunt de fitxers que conformen el treball pràctic, amb el producte desenvolupat. En la taula següent es detallen i descriuen cada un dels fitxers adjunts:

Fitxer	Descripció
llegiu-me.txt	Document on s'explica com instal·lar i utilitzar el producte desenvolupat, és a dir, els codis fonts i fitxers adjunts.
agarrigarov_presentació.pdf	Presentació, resum del treball
agarrigarov_TFG_script_1-taules-BD.sql	Script SQL per crear les taules de la BD
agarrigarov_TFG_script_2-log-BD.sql	Script SQL per crear la taula del diari o log de les accions fetes a la BD
agarrigarov_TFG_script_3-proc-BD.sql	Script SQL per carregar els procediments i funcions a la BD
agarrigarov_TFG_script_4-dades-BD.sql	Script SQL per carregar registres a la BD per poder efectuar proves
agarrigarov_TFG_script_5-taules-DWH.sql	Script SQL per crear les taules del DWH
agarrigarov_TFG_script_6-proc-DWH.sql	Script SQL per carregar els procediments al DWH
agarrigarov_TFG_script_7-proves-DWH.sql	Script SQL per carregar registres de proves al DWH
agarrigarov_TFG_script_8-extrac-BD.psql	Comandes 'psql' per extreure la informació de la BD que posteriorment carregarem al DWH.
agarrigarov_TFG_script_9-carregar-DWH.sql	Script SQL per carregar els registres extrets de la BD al DWH
agarrigarov_TFG_web.zip	Directorí 'UOC-TFG-BBDD', i tots els fitxers que en pengen, corresponent a l'aplicació web d' <i>Apache Tomcat</i> .

Taula A10.1: Relació de fitxers adjunts a la memòria del treball