

# **Comerç Electrònic amb J2EE**

**Josep Villar Azorín**

Enginyeria Tècnica d'Informàtica de Sistemes

**Consultor: Jordi Ceballos Villach**

18 de Juny de 2004

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

*A la Laura i el Jan. Aquest treball també és una mica d'ells.*

*A "Don José", que de ben segur li hauria agradat.*

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## RESUM

L'objectiu principal d'aquest Treball de Fi de Carrera ha estat, més que el que podria suggerir el seu títol, el disseny d'una infraestructura per a desenvolupar aplicacions de Comerç Electrònic amb tecnologia J2EE, tot i que es presenta una implementació específica com a *demonstració* de la seva viabilitat.

La idea essencial d'aquesta infraestructura rau en el concepte de la independència de la procedència de les dades respecte a la lògica pròpia del funcionament d'una *Botiga Virtual*. Aquesta independència no vol dir quin tipus o fabricant de base de dades es pot emprar tot utilitzant un SQL estàndard, si no que la possibilitat d'utilitzar diferents peces de software ja existent, aprofitant així els Sistemes d'Informació dels que ja disposa el client com poden ser un ERP, una base de dades pròpia, un sistema de fitxers propietari, un sistema transaccional basat en *Mainframe* (com per exemple CICS), uns *Web Services* que puguin facilitar els seus proveïdors, etc.

Per a obtenir aquesta independència, s'han definit diferents *Serveis* que proporcionen cadascuna de les necessitats que té l'esmentada infraestructura. Cada servei és totalment independent l'un de l'altre i pot tenir una procedència de dades diferent. Per exemple, podria existir una *Botiga Virtual* on tot el catàleg (*Categories, Productes...*) fos aliè al client (d'un proveïdor concret) i l'obtingués a partir d'un *Web Service* ofert pel proveïdor i, per una altra banda, la gestió d'usuaris (clients finals bàsicament) estigués implementada a partir d'una API facilitada pel fabricant de l'ERP que utilitza el propietari de la *Botiga Virtual*.

Un altre aspecte d'aquesta infraestructura és la utilització d'un *Framework* estàndard, *Jakarta Struts*, per a la implementació del que és l'aplicació *Web* pròpiament dita.

La implementació final lliurada està suportada per una base de dades MySQL dissenyada a tal efecte.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## Índex de continguts

RESUM .....	3
<b>1. INTRODUCCIÓ.....</b>	<b>5</b>
1.1. PUNT DE PARTIDA .....	5
1.2. OBJECTIUS .....	7
1.3. ENFOCAMENT I MÈTODE SEGUIT .....	7
1.4. PLANIFICACIÓ DEL PROJECTE .....	9
1.5. PRODUCTES OBTINGUTS .....	11
<b>2. INFRASTRUCTURA PER AL DESENVOLUPAMENT .....</b>	<b>12</b>
2.1. MODEL .....	15
2.1.1. Beans .....	15
2.1.2. Serveis .....	18
2.1.3. Implementació específica (MySQL).....	20
2.2. VISTA.....	22
2.3. CONTROLADOR.....	23
<b>3. APLICACIÓ DE COMERÇ ELECTRÒNIC.....</b>	<b>24</b>
3.1. WEB PÚBLICA .....	24
3.2. BOTIGA VIRTUAL.....	26
3.3. EINA D'ADMINISTRACIÓ.....	29
<b>4. CONCLUSIONS.....</b>	<b>44</b>
<b>5. BIBLIOGRAFIA .....</b>	<b>46</b>

## Índex de figures

Figura 1. Planificació.....	10
Figura 2. Arquitectura de Model 1 .....	13
Figura 3. Arquitectura de Model 2 .....	14
Figura 4. Diagrama E/R de la implementació lliurada .....	21
Figura 5. Estructura de l'aplicació.....	22
Figura 6. Menú de la Web Pública .....	25
Figura 7. Menú de la Botiga Virtual.....	26
Figura 8. Cistella de la compra .....	27
Figura 9. Fitxa de producte.....	28
Figura 10. Menú d'administració .....	29
Figura 11. Manteniment de categories .....	30
Figura 12. Relació de productes d'una categoria .....	31
Figura 13. Pantalla de detall d'un producte .....	32
Figura 14. Manteniment de fitxa tècnica.....	33
Figura 15. Manteniment de relació de recanvis d'un producte .....	34
Figura 16. Relació de marques (fabricants) .....	35
Figura 17. Relació de models d'una marca.....	36
Figura 18. Relació de recanvis (adaptables) d'un model .....	37
Figura 19. Manteniment de característiques .....	38
Figura 20. Clients - Cerca .....	39
Figura 21. Clients - Detall .....	40
Figura 22. Comandes - Cerca .....	41
Figura 23. Comandes - Detall sense modificació .....	42
Figura 24. Comandes - Detall amb modificació .....	43

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

# 1. Introducció

## 1.1. Punt de partida

El punt de partida d'aquest projecte és l'obtenció d'una aplicació de comerç electrònic per a un fabricant de motocicletes (a partir d'ara *fabricant*). El *fabricant* comercialitza tant productes propis com productes de tercers, no sent significativa la procedència del producte.

L'eix central d'aquesta aplicació és el *catàleg*. El catàleg és el conjunt de productes, tant propis com aliens, que oferta el *fabricant*. Aquest conjunt de productes estan organitzats en una jerarquia de categories i subcategories disposades en una estructura d'arbre. A efectes pràctics només es consideren 2 nivells, els quals anomenarem *categories* i *subcategories*. Un producte, al marge d'atributs genèrics tals com descripció, referència, imatge, etc., pot disposar del que s'anomena la *fitxa tècnica*, que no és més que un conjunt de possibles característiques del producte amb els seus valors corresponents (pot tenir sentit parlar del diàmetre d'un cilindre però no pas del diàmetre d'un cargol quadrat). Per una altra banda els productes poden tenir entre ells relacions de composició / recanvi, és a dir, un producte pot estar compost d'altres o un producte pot ser recanvi d'altres (o cap de les dues possibilitats). Tanmateix, qualsevol producte es pot considerar com a un recanvi o adaptable d'un determinat model de motocicleta. Els models de motocicleta, per la seva part, pertanyen a les diferents marques de motocicleta que hi ha al mercat.

Aquesta aplicació està dividida en tres parts diferenciades:

- Una eina d'administració que permetrà el manteniment i el control de les dades relacionades amb l'aplicació.
- Una *Web Pública* (a partir d'ara *WP*) on qualsevol usuari pugui consultar el catàleg de productes i les seves característiques des d'Internet. Aquesta consulta es podrà fer tant buscant per categories / subcategories com per models d'una marca de motocicleta concreta.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

- La *Botiga Virtual* (des d'ara *botiga*) on un usuari registrat podrà compondre una comanda a partir de productes oferts des de la *WP* i cursar aquesta comanda al *fabricant*.

A partir d'aquestes tres parts, es pot establir el mateix número de rols implicats en l'aplicació:

- Un usuari *administrador*, que faria ús de l'eina d'administració.
- Un usuari *visitant*, el qual podria fer ús de la *WP* per consultar el catàleg del *fabricant*.
- Un usuari *client*, que es tractaria d'una especialització de l'usuari *visitant*, però que un cop identificat com a client, podria compondre, emmagatzemar i/o cursar una comanda.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## 1.2. Objectius

Tal i com s'ha comentat en el resum, l'objectiu principal d'aquest TFC és la implementació d'una infraestructura per a la construcció d'aplicacions de Comerç Electrònic basades en J2EE que permetin la llibertat total de l'elecció de la procedència dels objectes de negoci propis de cada instal·lació.

Aquest objectiu principal està complementat amb el disseny i desenvolupament d'una aplicació concreta de Comerç Electrònic, una *Botiga Virtual*, per a un fabricant de recanvis per a motocicletes.

Aquesta aplicació concreta servirà com a *demonstració* de la viabilitat de la infraestructura proposada en el present treball.

Per una altra banda, també s'ha considerat com a objectiu que les aplicacions construïdes a partir d'aquesta infraestructura tinguessin la característica d'acceptar diferents idiomes.

Un altre objectiu, secundari, que s'havia establert era l'obtenció d'un producte compatible amb qualsevol plataforma J2EE existent al mercat. La utilització d'una implementació estàndard d'un contenidor *Web* com és el cas de *Jakarta Tomcat* i el seguiment de l'especificació *Servlets 2.3*, ha assegurat aquest objectiu.

## 1.3. Enfocament i mètode seguit

L'enfocament d'aquest projecte és resultat d'una fase prèvia d'estudi de diferents botigues virtuals, fent un recull de les seves característiques i necessitats.

A partir d'aquest recull s'ha fet una agrupació de les diferents *entitats* i *accions* que eren comuns entre ells mateixos, obtenint així el concepte de *Servei* i, finalment, la relació dels *Serveis* necessaris per a la implementació d'una *Botiga Virtual* tipus.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Per una altra banda, en aquest recull es van identificar les diferents entitats (objectes de negoci) que intervenien en les diferents *Botigues Virtuals* estudiades, així com les necessitats del *client final*.

Tota aquesta part va ser desenvolupada a partir de l'Anàlisi Funcional lliurat.

Cada *Servei* proporciona la manipulació necessària dels objectes de negoci que formen la *Botiga Virtual*. De fet, cada *Servei* ofereix els diferents casos d'ús relacionats amb cadascun d'aquests objectes de negoci.

Un cop obtinguda la relació definitiva dels *Serveis* i les seves interaccions (casos d'ús) amb els diferents objectes de negoci es van dissenyar les tres parts que consta la *Botiga Virtual* lliurada:

- Eina d'administració
- *Web Pública*
- Gestió de clients i comandes (*Botiga virtual* pròpiament dita)

El *Document de Disseny*, tenint com a punt de partida l'*Anàlisi Funcional*, va ser el document resultant d'aquesta tasca.

Finalitzat el disseny, es va procedir amb la implementació tot seguint un cicle de vida iteratiu.



Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

#### 1.4. Planificació del Projecte

A efectes pràctics la planificació del projecte va ser la següent:

- **Document de definició**

Elaboració d'un document de definició on es presentava el projecte després d'una fase inicial d'estudi de diferents *Botigues Virtuals*.

- **Anàlisi**

La fase d'anàlisi va tenir 3 setmanes de durada. Es va dividir en 2 subfases:

- Recollida de requisits (entrevista amb l'usuari) i profundiment en l'estudi d'altres aplicacions similars
- Elaboració de l'anàlisi funcional.

- **Disseny**

Un cop obtingut l'anàlisi es procedirà al disseny tècnic de l'aplicació. Es concreta el model de dades i de classes, així com un detall de cada *Servei* disponible.

- **Desenvolupament**

Amb el disseny elaborat comença la fase de desenvolupament a la que s'han 6 setmanes.

- **Memòria**

Elaboració del present document

- **Confecció de la presentació**

Lliurament final, que consta d'una presentació juntament amb present document i el codi de l'aplicació.

A continuació es mostra el diagrama de *Gantt* amb la planificació descrita:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

# Planificació del Projecte - Diagrama de Gantt

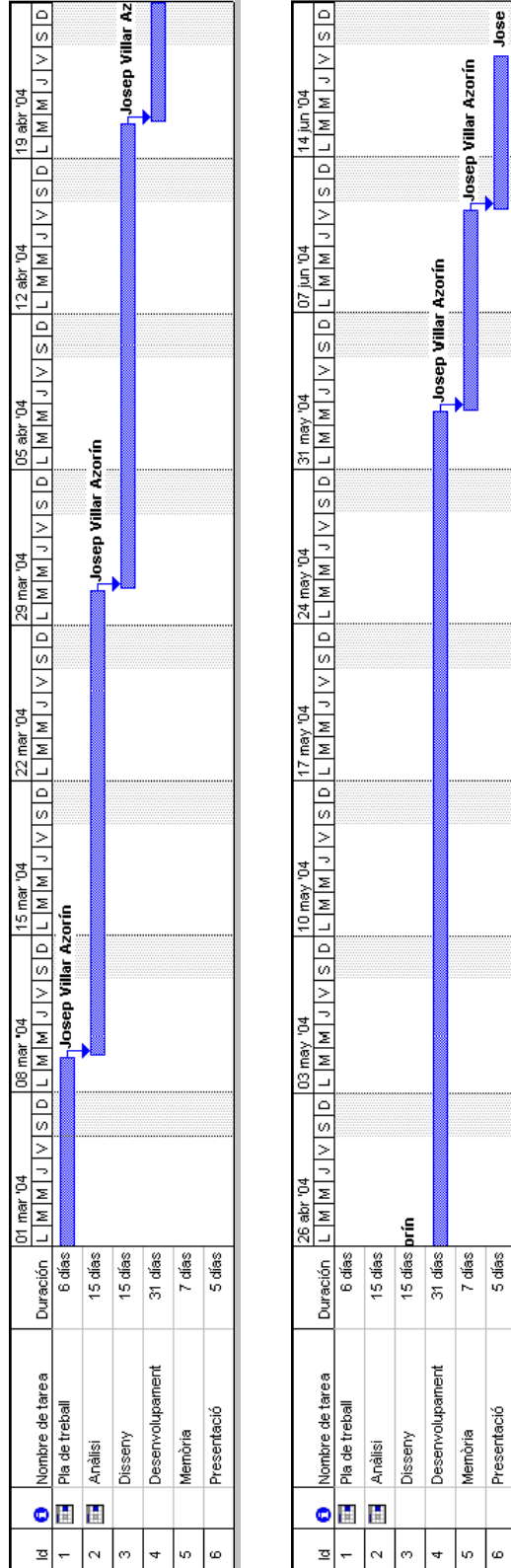


Figura 1. Planificació.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

### 1.5. Productes obtinguts

Els productes obtinguts amb aquest projecte han estat els proposats:

- Un conjunt de classes – objectes de negoci i *serveis* – que conformen la infraestructura bàsica per a desenvolupar, per una banda.
1. Una aplicació de comerç electrònic amb una implementació dels serveis específica, basada en MySQL. Aquesta aplicació es divideix en 3 subaplicacions:
    - 1.1. Administració
    - 1.2. *Web pública*
    - 1.3. *Botiga Virtual*

En els següents capítols del present document es detallaran cadascun dels productes obtinguts.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## 2. Infraestructura per al desenvolupament

En aquest capítol es descriu tota la infraestructura dissenyada per a desenvolupar aplicacions de comerç electrònic independents de la procedència de les dades. Tanmateix, s'expliquen les diferents eines i tecnologies utilitzades per al seu desenvolupament.

Per experiència el desenvolupament d'aplicacions basades en J2EE que combinen HTML, JSPs, codi Java, etc., és difícil de dur a terme (i encara més difícil el manteniment) sense un disseny flexible i mantenible.

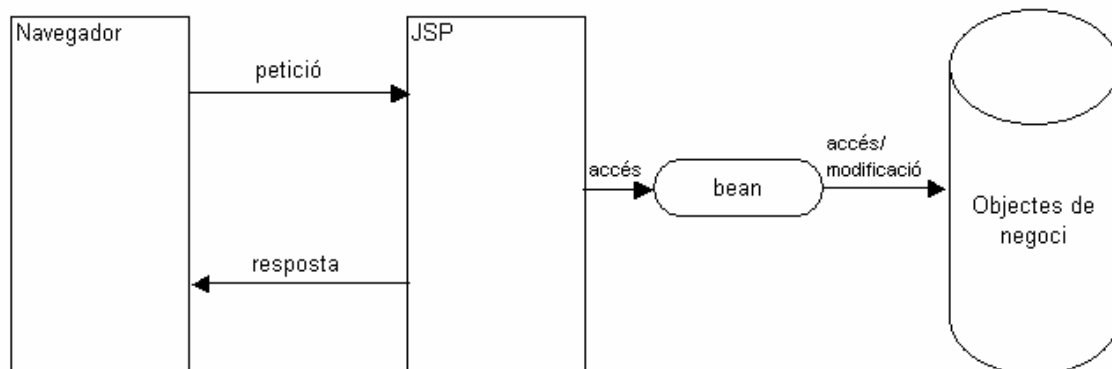
Dintre de la tecnologia triada per al desenvolupament d'aquesta aplicació (J2EE) es poden trobar les següents aproximacions de disseny:

- Fer que les JSP s'encarreguin directament de tant la presentació de les dades, com de l'obtenció d'aquestes així com d'establir els diferents fluxos de navegació de l'aplicació.
- Focalitzar només la presentació en JSP i deixar en mans de *beans* la resta de la funcionalitat. Aquesta aproximació és coneguda com arquitectura de Model 1.
- Separar la presentació, l'obtenció de dades i el control del flux de l'aplicació en capes i delegar en JSPs, *beans* i *servlets* les diferents funcionalitats corresponents a les capes per a implementar una arquitectura MVC (*Model-View-Controller*), coneguda també com arquitectura de Model 2.

La primera aproximació no aporta absolutament res, tret de codi confús, no reaprofitable i de molt difícil manteniment i escalabilitat per lo qual és totalment descartable.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

L'arquitectura de Model 1 ja representa un a millora respecte a l'anterior. La seva organització es pot veure en el següent gràfic:



**Figura 2. Arquitectura de Model 1**

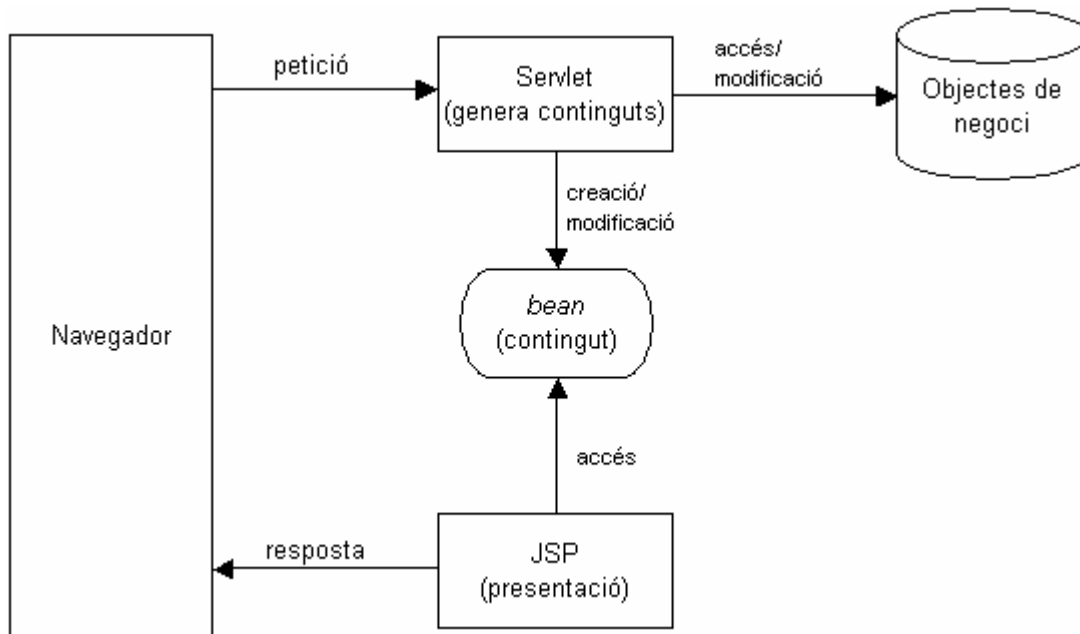
Les pàgines JSP accedeixen als objectes de negoci indirectament mitjançant els *beans*. Amb això s'està aïllant a les JSP dels canvis produïts als objectes de negoci (si no es canvia la definició dels *beans*).

El problema radica en què les pàgines JSP també són les responsables de generar el contingut, és a dir, d'instanciar els *beans* apropiats per a la seva presentació. Per fer-ho, el desenvolupador ha d'emprar Java dintre de les JSP (cosa que, encara que impossible, s'ha d'evitar al màxim). Aquest fet porta a que els desenvolupadors de software es veuen implicats tant en el desenvolupament d'objectes de negoci com en la generació de la presentació. Això incrementa la dificultat per aconseguir una divisió dintre de l'equip de treball (desenvolupadors de software i autors de planes). Aquesta divisió en projectes grans és molt important.

Finalment, l'arquitectura Model 2 produeix un resultat fàcil de mantenir i d'extendre. A l'igual que en l'arquitectura de Model 1 separa els objectes de negoci de la presentació, però a més, separa la generació del contingut de la presentació.

En el següent gràfic es pot veure un esquema de l'arquitectura Model 2:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes



**Figura 3. Arquitectura de Model 2**

En aquesta arquitectura les peticions són enviades a un *servlet* que accedeix als objectes de negoci per crear el contingut. Aquest contingut s'emmagatzema en un *bean* (patró de disseny *Value Object* o *Data Transfer Object*), el qual és accedit per una pàgina JSP que s'encarrega de la seva presentació. A més, el *servlet* s'encarrega del control del flux de l'aplicació.

El Model 2 és una arquitectura MVC on els objectes de negoci representen el **M**odel, les pàgines JSP representen les **V**istes (presentació) i els *servlets* representen els **C**ontroladors (i la generació dels continguts).

L'arquitectura de Model 2 serà la que s'utilitzarà per a la construcció de l'aplicació de Comerç Electrònic que ens ocupa.

Hi ha diversos *frameworks* ja desenvolupats que implementen arquitectures de Model 2:

- Jakarta Struts
- Cocoon

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

- Expresso
- Freemarker
- Velocity
- WebMacro
- Maverick
- SiteMesh
- Jakarta Turbine
- WebWork
- JavaServer Faces

De tots aquests el més estès i documentat és Jakarta Struts per lo qual ha estat l'utilitzat per a dur a terme el projecte. Entre les altres opcions hi ha, però, d'algunes que poden utilitzar-se en combinació amb Struts com és el cas de JavaServer Faces.

A continuació es detalla cadascuna de les capes que conformen la infraestructura descrita.

## 2.1. Model

La capa *Model* està diferenciada en 3 parts:

- Els diferents *beans* utilitats en l'aplicació
- El conjunt de *Serveis* definits (Interfaces)
- La implementació específica dels *Serveis*

### 2.1.1. Beans

Els *beans* són el mitjà de comunicació de dades dintre de tota l'arquitectura. Tècnicament no són més que simples classes *Java* amb una sèrie d'atributs i els seus corresponents mètodes *getter* i *setter*. No tenen cap mena de lògica de negoci associada. Simplement són contenidors de dades.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Aquestes classes són candidates a ser passades pel *request* dintre d'altres estructures de dades com ara diferents implementacions *java.util.List*, *java.util.Map*, etc., així com són susceptibles de ser emmagatzemades (com per exemple la comanda actual, les dades de l'usuari en curs...) a la sessió. És per aquesta raó que tots els *beans* definits han d'implementar l'*interface java.io.Serializable*. Si es vol *escalar* l'aplicació i tenir-la en un servidor d'aplicacions en *cluster* amb sessió distribuïda, serà requisit indispensable que tots els objectes que s'emmagatzemin a la sessió implementin aquesta *interface*.

S'han proposat, com a resultat de l'estudi previ d'altres aplicacions de comerç electrònic, així com de les conclusions obtingudes arrel de les entrevistes amb el *client* final, una sèrie de *beans* com a part base de la infraestructura. Aquests *beans*, però, poden ser diferents depenent de la implementació final, sempre i quan es respecti el seu nom original.

Aquests *beans* són:

- CategoryVO  
Conté la informació referent a les categories en què està organitzat el catàleg de productes. Es tracta d'una relació jeràrquica on cada categoria té el seu *pare*. És un dels elements que té un tractament per a diferents idiomes i per aquest motiu té diversos mètodes d'obtenció de determinats atributs depenents de l'idioma (com ara descripció, imatge...)
- CountryVO  
Informació bàsica de països. Un país té un identificador intern, un identificador *normalitzat* (ISO) i el seu nom.
- FeatureVO  
Característiques atribuïbles a un producte concret. Amb tractament per a diferents idiomes.
- ItemVO



Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Informació bàsica d'un producte. Amb tractament multi-idioma (descripcions...).

- ManufacturerVO  
Fabricant d'on els seus productes fan ús dels productes de la *botiga virtual* (és a dir, els productes de la *botiga virtual* són *recanvis* o *complements* dels *models* que ofereix aquest fabricant).
- ModelVO  
Model d'un producte d'un fabricant donat, el qual és susceptible de tenir com a *recanvi* un dels productes del catàleg.
- OrderLineVO  
Informació bàsica d'una línia de comanda: producte, quantitat, preu...
- OrderVO  
Comanda. Conté informació com ara identificador de client de la comanda, data, número de comanda...
- TechnicalCardVO  
Característica específica d'un producte del catàleg, a partir d'una *Feature* (veure *FeatureVO*) i d'un valor concret. Amb suport multi-idioma.
- UserVO  
Informació relativa a un usuari de l'aplicació, ja sigui *client* o *administrador*, depenent del seu *rol*.

Tots aquest *beans* estan estretament relacionats amb els diferents *Serveis* oferts per la infraestructura.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

### 2.1.2. Serveis

En aquest projecte, un dels requeriments principals és la independència dels objectes de negoci respecte a l'aplicació final. Es pretén que l'aplicació sigui configurable de tal forma que l'origen de les dades pugui ser diferent no tan sols en diferents instal·lacions si no també dintre d'una mateixa instal·lació. És a dir, les dades dels clients podrien estar en un Mainframe i accedir a elles mitjançant transaccions CICS™ i les dades referents a productes podrien formar part d'un ERP aliè a l'aplicació però del qual tenim un mitjà de comunicació amb aquest (bé sigui coneixent l'estructura de la base de dades o mitjançant altres mètodes com RMI, CORBA, WebServices...).

A tal efecte es definiran el que anomenarem *serveis* per tal de gestionar els diferents objectes de negoci. Cada *servei* tindrà associada una *interface* i, per lo tant, podran existir diferents implementacions d'aquestes *interfaces*.

Els diferents *serveis* necessaris per a dur a terme el projecte són els següents:

- *CategoryService*  
Gestió de la jerarquia de categories de productes.
  
- *ManufacturerService*  
Gestió de fabricants i models
  
- *UserService*  
Gestió d'usuaris (tant administradors com clients)
  
- *OrderService*  
Gestió de comandes. Inclou la gestió de la *cistella* de la compra.
  
- *ItemService*  
Gestió de productes. Donada la complexitat del que envolta a un producte (fitxa tècnica, característiques, recanvis, adaptables...) s'ha separat de la gestió de categories.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

- *FeatureService*  
Gestió de característiques.
  
- *CountryService*  
Gestió de països.
  
- *LanguageService*  
Gestió d'idiomes.

Els dos últims *serveis* són auxiliars i només tenen mètodes d'accés a la llista dels diferents elements possibles.

Les implementacions s'obtiniran a partir de crear una instància amb el mètode

```
InterfaceService x = (InterfaceService) Class.forName(serviceName).newInstance
```

Cada *servei* té assignat un identificador (*serviceName*). A partir d'aquest identificador es pot obtenir el nom de la classe que implementa la *Interface* en concret. Aquesta relació identificador-nom de classe es troba en la parametrització de l'aplicació *Web* ubicada en el fitxer `/aplicacio/WEB-INF/web.xml` (on *aplicacio* és el context-root de l'aplicació implementada). S'aprofita la pròpia parametrització (`init-param`) del *servlet* emprat per a emmagatzemar aquests valors:

```
<servlet>
  <servlet-name>tfc</servlet-name>
  <servlet-class>
    org.apache.struts.action.ActionServlet
  </servlet-class>
  <init-param>
    <param-name>categoryservice</param-name>
    <param-value>
      com.uoc.tfc.mysql.CategoryServiceImpl
    </param-value>
  </init-param>
  .
  .
  .
```

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

### 2.1.3. Implementació específica (MySQL)

Per a la implementació específica d'aquest projecte s'ha utilitzat una base de dades MySQL i s'han desenvolupat les corresponents implementacions de cada *servei* definit a la infraestructura:

Servei	Implementació
<code>com.uoc.tfc.services.CategoryService</code>	<code>com.uoc.tfc.mysql.CategoryServiceImpl</code>
<code>com.uoc.tfc.services.CountryService</code>	<code>com.uoc.tfc.mysql.CountryServiceImpl</code>
<code>com.uoc.tfc.services.FeatureService</code>	<code>com.uoc.tfc.mysql.FeatureServiceImpl</code>
<code>com.uoc.tfc.services.ItemService</code>	<code>com.uoc.tfc.mysql.ItemServiceImpl</code>
<code>com.uoc.tfc.services.LanguageService</code>	<code>com.uoc.tfc.mysql.LanguageServiceImpl</code>
<code>com.uoc.tfc.services.ManufacturerService</code>	<code>com.uoc.tfc.mysql.ManufacturerServiceImpl</code>
<code>com.uoc.tfc.services.OrderService</code>	<code>com.uoc.tfc.mysql.OrderServiceImpl</code>
<code>com.uoc.tfc.services.UserService</code>	<code>com.uoc.tfc.mysql.UserServiceImpl</code>

Cadascuna d'aquestes implementacions hereten de la classe `com.uoc.tfc.mysql.FeatureServiceImpl`, que proporciona una sèrie de mètodes comuns (obtenció de la connexió, etc...).

S'han utilitzat les classes que proporciona el mateix MySQL per a la connexió des de l'aplicació amb la base de dades.

Aprofitant que el *Framework Struts* permet definir un *pool* de connexions, aquest és el que s'ha utilitzat en comptes del que pot oferir el mateix *Tomcat*, per facilitar la instal·lació de l'aplicació (tenir un *pool* de connexions configurat a nivell de *Tomcat* no només requereix parametrització al *web.xml*, si no que a més es requereix una parametrització addicional en el fitxer de configuració de *Tomcat*, *server.xml*, dificultant així la distribució del producte. Com que la configuració de *Struts* (*struts-config.xml*) està dintre de l'aplicació, la distribució és més senzilla).

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Per a aconseguir el model de negoci basat en els *beans* descrits, el disseny de la base de dades va ser el següent:

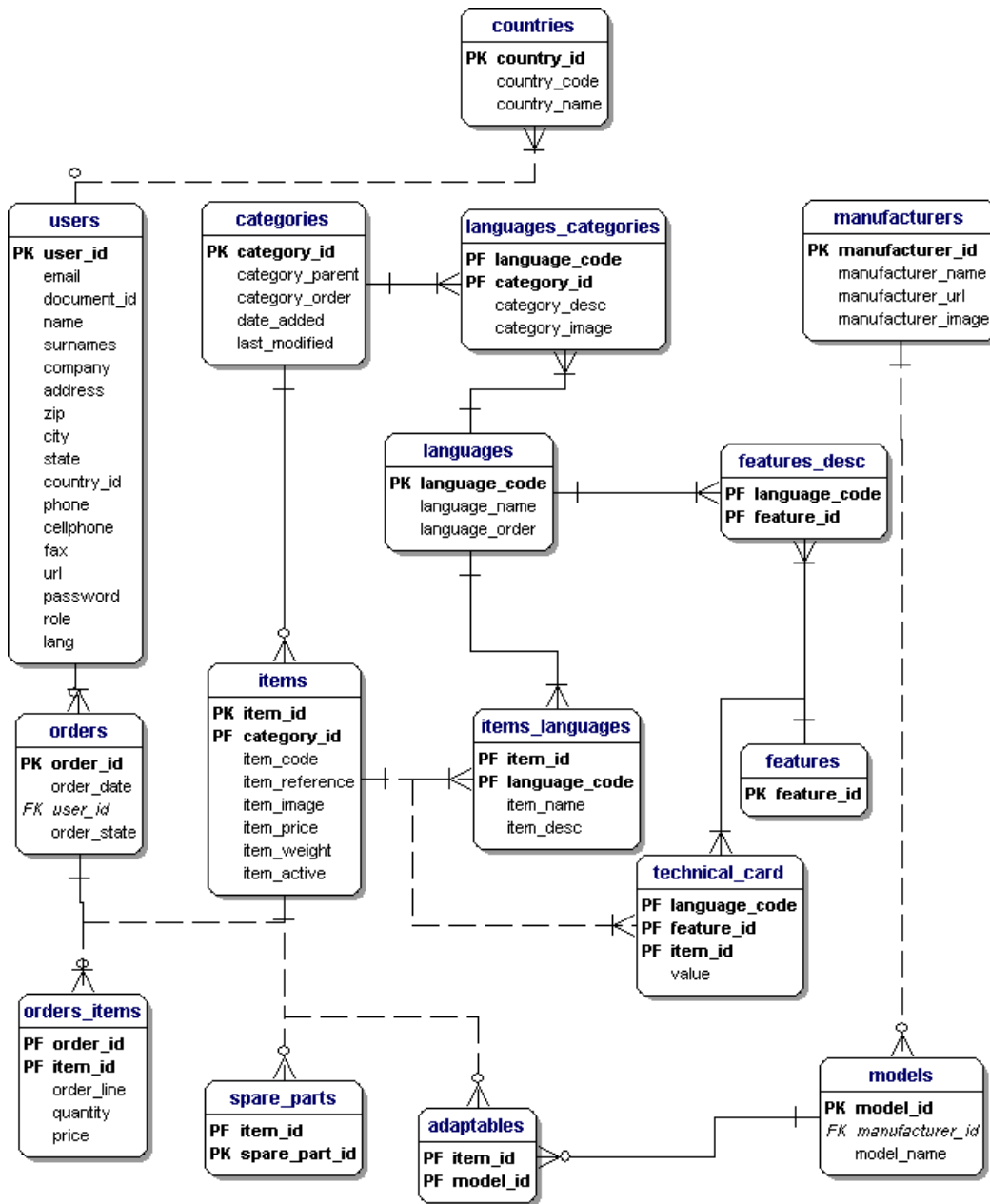


Figura 4. Diagrama E/R de la implementació lliurada

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## 2.2. Vista

Fent servir un contenidor *Web* la capa de *Vista (View)* s'ha implementat amb *Java Server Pages (JSP)*.

La disposició de les pantalles de l'aplicació té la següent estructura:

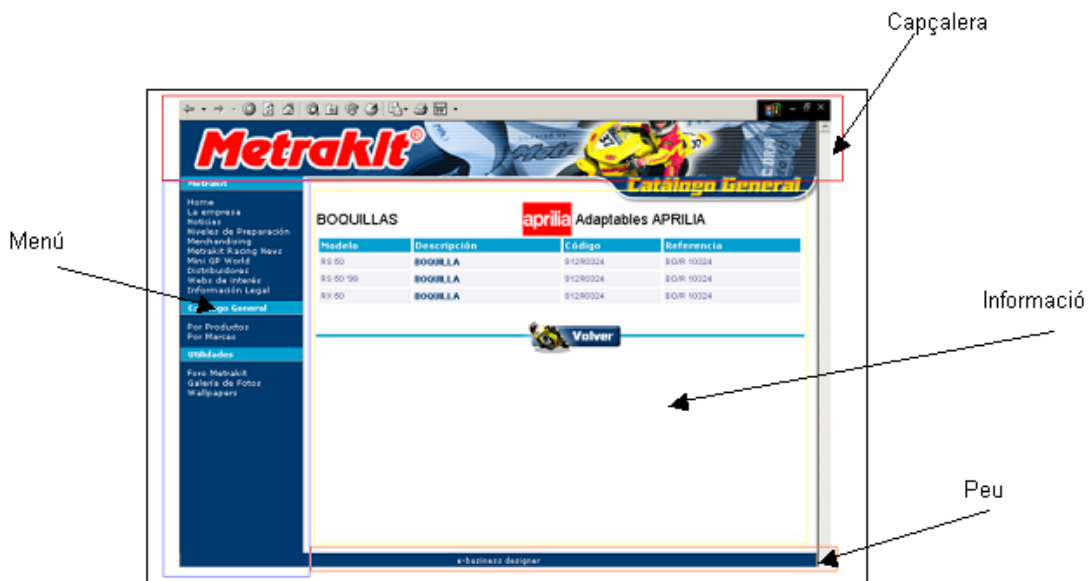


Figura 5. Estructura de l'aplicació.

Per a la seva realització, una solució seria construir un *frameset* i anar canviant el *frame* corresponent a la zona d'informació, però, per experiència, aquesta solució porta bastants problemes. Tampoc seria bona solució que totes les pàgines continguessin el codi relatiu a totes les zones de la pantalla. Seria pràcticament immantenible.

La solució adoptada al problema és la d'utilitzar una llibreria de *tags* que incorpora *Struts*, anomenada *tiles*. Amb aquesta llibreria es poden definir *templates* que divideixen la pantalla en *regions* i on a cada regió se li pot assignar un *JSP*.

El mecanisme és el següent:

- Es defineix un *template* (o més, depenent de les diferents tipologies de pantalles)

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

- Es crea una JSP amb la pantalla a implementar que inclogui aquest *template*.

A cada regió de la plantilla se li assignen les JSP corresponents al menú, capçalera, peu (aquestes normalment sempre seran les mateixes) i informació.

Finalment, notar que s'ha fet ús dels fulls d'estil i de la codificació del codi seguint l'estàndard XHTML.

### 2.3. Controlador

Donat que s'utilitza el *framework Struts* per desenvolupar l'aplicació, aquesta és la part més dependent d'aquest *framework*.

L'aplicació té un *servlet* que fa de controlador i és propi de *Struts*:

```
org.apache.struts.action.ActionServlet
```

Es faran servir 3 mappings diferents per cridar-lo:

/wp/*	<i>Web Pública</i>
/manager/*	Administració
/shopping/*	<i>Botiga Virtual</i>

Per una altra banda, cadascun dels casos d'ús descrits al document d'anàlisi té associat una classe que s'encarregarà de la generació de continguts i control de flux.

Cadascuna d'aquestes classes associades heretarà d'una classe comuna. Aquesta classe comuna, a la seva vegada, heretarà de la classe `org.apache.struts.action.Action`. La raó d'aquesta classe *intermitja* és tenir un *punt comú* (mètode `getService()`) a totes les classes *Action* de l'aplicació per tal de fer tasques comunes, com l'obtenció del servei apropiat per a cada classe *Action*, assegurar-se que l'usuari es vàlid, gestió de l'idioma...

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## 3. Aplicació de Comerç Electrònic

En aquest document es descriu de forma detallada cadascuna de les parts que componen l'aplicació de comerç electrònic continguda en el projecte.

Tal i com ja s'ha avançat, l'aplicació de comerç electrònic exposada es divideix en:

- *Web Pública*
- *Botiga Virtual*
- *Administració*

### 3.1. Web Pública

La Web Pública és la part de l'aplicació que és accessible des d'Internet des de qualsevol lloc i per qualsevol usuari (registrat o no). La *URL* d'accés és *http://servidor/tfc/home*.

Les funcionalitats disponibles en aquesta part de l'aplicació són:

- Consulta del catàleg
  - Per Productes

Un usuari selecciona una categoria de les disponibles, i se li presenta la llista de les diferents marques de fabricants de motocicletes disponibles. Un cop triada una marca, es mostra la relació de models d'aquesta marca, i en escollir un model concret, es presenta la llista de productes relacionats amb el model. Finalment, en elegir un producte, surt la fitxa tècnica del producte en qüestió.



Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

- Per Marques

Un usuari selecciona una marca determinada, i li apareix la llista de models que pertanyen a la marca. Un cop triat un model, surt la llista de productes relacionats amb el model, ordenats per categoria de producte. Finalment, en elegir un producte, surt la fitxa tècnica del producte en qüestió.

- Identificació

Un usuari registrat prèviament pot identificar-se com a tal. A tal efecte, se li presenta una finestra on se li demana el seu e-mail (identificador) i una clau d'accés. Si l'accés és correcte, s'obre una nova sessió de compra.

- Registre

Funcionalitat que permet a un usuari no registrat donar-se d'alta al sistema. L'usuari ha d'introduir un seguit de dades obligatòries i, després d'una sèrie de validacions, es dona d'alta i se li obre una nova sessió de compra.

Totes aquestes funcionalitats són accessible des del menú que apareix a la part esquerra de la pantalla:

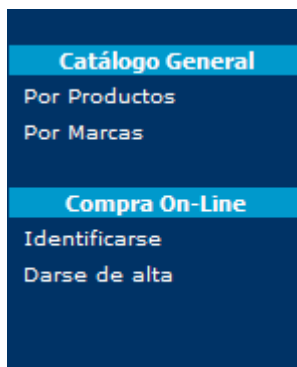


Figura 6. Menú de la *Web Pública*

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

### 3.2. Botiga Virtual

Tal com s'ha vist, des de la *Web Pública* un usuari es pot identificar i accedir a les funcionalitats que ofereix la *Botiga Virtual*.

L'usuari manté, a més, les funcionalitats de consulta de catàleg de la *Web Pública*.  
..

El menú de l'esquerra canvia es presenta les següents opcions:

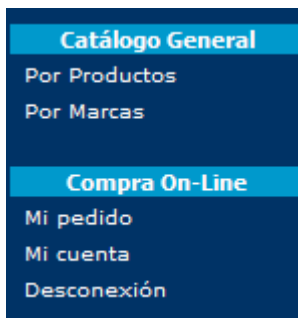


Figura 7. Menú de la *Botiga Virtual*.

Tot seguit es presenten les funcionalitats pròpies de la *Botiga Virtual*:

- Gestió de la comanda en curs del client  
Aquesta opció és accessible directament des del menú de l'esquerra. Un client pot consultar la seva comanda actual, modificar les quantitats dels productes que la componen (que prèviament ha anat afegint el client), i formalitzar la comanda. A tal efecte se li presenta la finestra de comanda:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

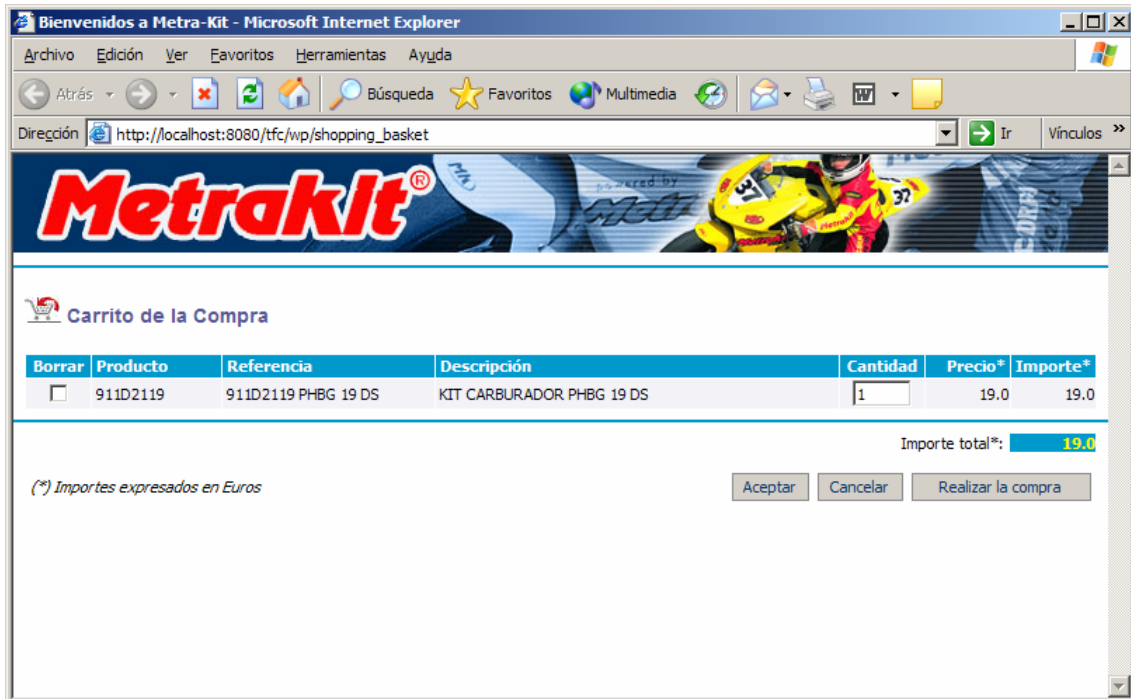


Figura 8. Cistella de la compra

L'opció *Aceptar* confirma les modificacions efectuades i no tanca la finestra. L'opció *Cancelar* descarta qualsevol canvi i tanca la finestra. També, tot triant en les caselles de verificació de la columna *Esborrar*, l'usuari pot donar de baixa línies de la comanda. Finalment, l'opció *Realitzar la compra*, demana la confirmació a l'usuari i *processa* la compra, és a dir, li canvia l'estat (inicialment estava en estat *Confecció per part del client*), passant-la a estat *Rebuda* (estat que és controlat per l'administrador). A partir d'aquest moment, l'usuari ja podria cursar una nova comanda. Un usuari, tot i que es desconnecti de l'aplicació, sempre manté la comanda en curs i la recupera en tornar-se a identificar.

- Consulta/Manteniment de les dades personals del client  
En aquesta opció es mostra al client la pantalla de manteniment d'usuaris, on aquest podrà modificar les seves dades personals. També és accessible des de el menú de l'esquerra de l'aplicació.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

### Desconnexió de la *Botiga Virtual*

Desconnecta a l'usuari de l'aplicació. Esborra de la sessió les dades referents a l'usuari (dades personals i comanda en curs) i recarrega la pàgina principal de la *Web Pública*.

També és accessible des de el menú de l'esquerra de l'aplicació.

- Afegir un producte a la comanda

A diferència de les anteriors, aquesta funcionalitat no és accessible des de el menú de la part esquerra de l'aplicació. Quan un usuari consulta el catàleg i arriba a la *Fitxa Tècnica* del producte, apareix **només quan l'usuari està identificat** un *link* que mostra la finestra de gestió de comandes ja descrita:



Descripción	Código	Referencia
KIT CARBURADOR SHA 16.16 C	91112150	91112150 SHA 16.16 C

ø adap. Filtro: 58.5  
 Tobera: brida  
 ø Colector: 19  
 Starter: cable  
 Depresor: no  
 Engrase: normal  
 Precio: 50.0 EUR

Comprar este producto

NOMÉS APAREIX QUAN L'USUARI ESTÀ IDENTIFICAT

Figura 9. Fitxa de producte

Si el producte no existia dintre de la comanda, el sistema afegeix una nova línia amb el producte triat i amb quantitat per defecte '1'. El cursor

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

es situa en la caixa de text per introduir la quantitat del producte seleccionat (ja sigui nou o no) i marca en un altre color (vermell en negreta) el seu codi, la seva referència i la seva descripció perquè l'usuari sigui conscient del que ha triat.

### 3.3. Eina d'Administració

Amb la URL <http://servidor/tfc/adm>, l'usuari accedeix a l'eina d'administració on, inicialment, només té l'opció per identificar-se com a administrador. En la implementació lliurada un usuari administrador és un usuari on el seu atribut *rol* l'identifica com a tal (A=Administrador / C=Client).

Un cop identificat l'administrador, l'aplicació li presenta, com sempre, a l'esquerra les diferents opcions de menú disponibles:

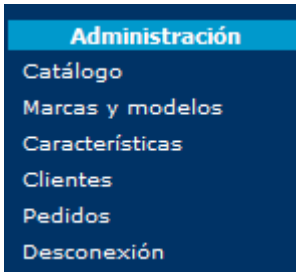


Figura 10. Menú d'administració

Tot seguit es detalla cadascuna d'elles:

#### 3.3.1. Catàleg

Des d'aquesta opció es manté el catàleg de productes de la *Botiga Virtual*. Tal i com ja s'ha descrit, el catàleg està organitzat per categories i subcategories. En la present aplicació només es consideren 2 nivells, i només en el segon nivell és on s'assignen els productes.

Un cop l'usuari tria aquesta opció de menú, apareix la llista de categories de primer nivell. Si es fa *click* sobre la descripció d'una d'elles, aquesta apareix d'altre color i es mostra informació relativa a ella a la part dreta de la pantalla:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

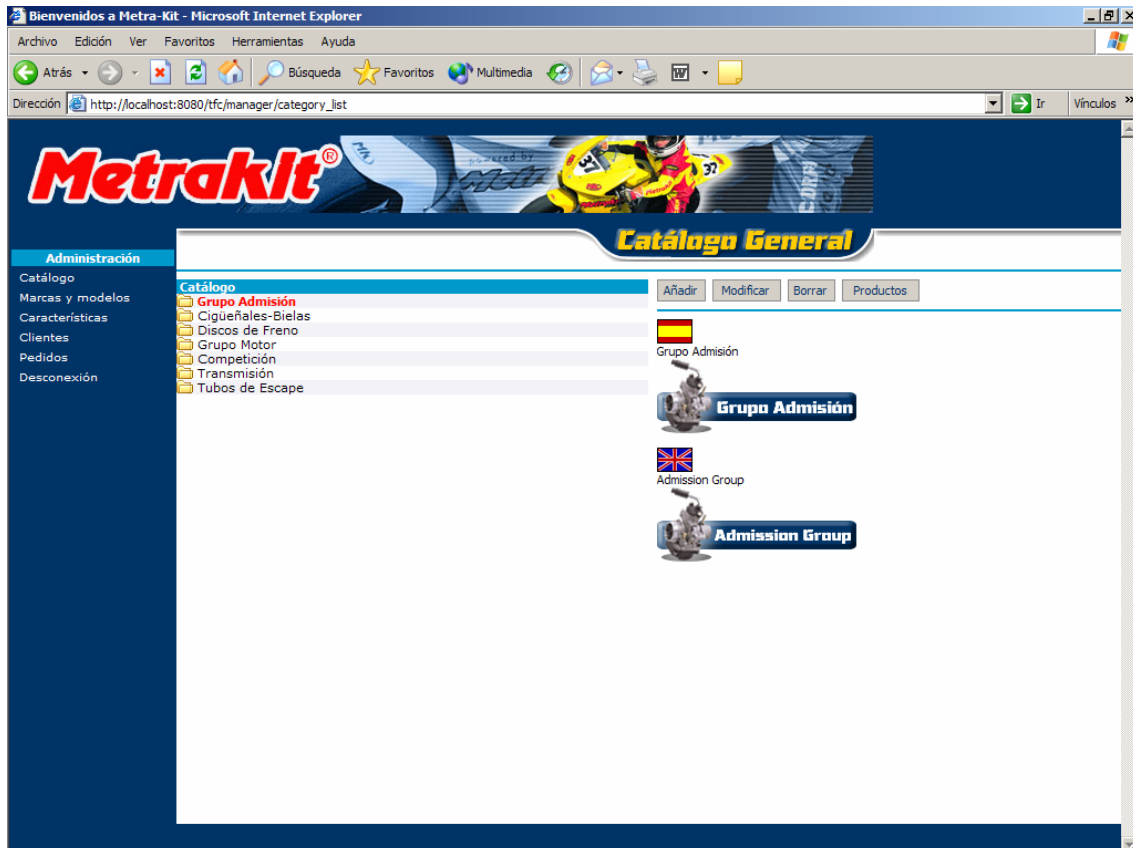


Figura 11. Manteniment de categories

Si es fa *click* sobre la imatge 📁 , es mostra la llista de categories *filles*.

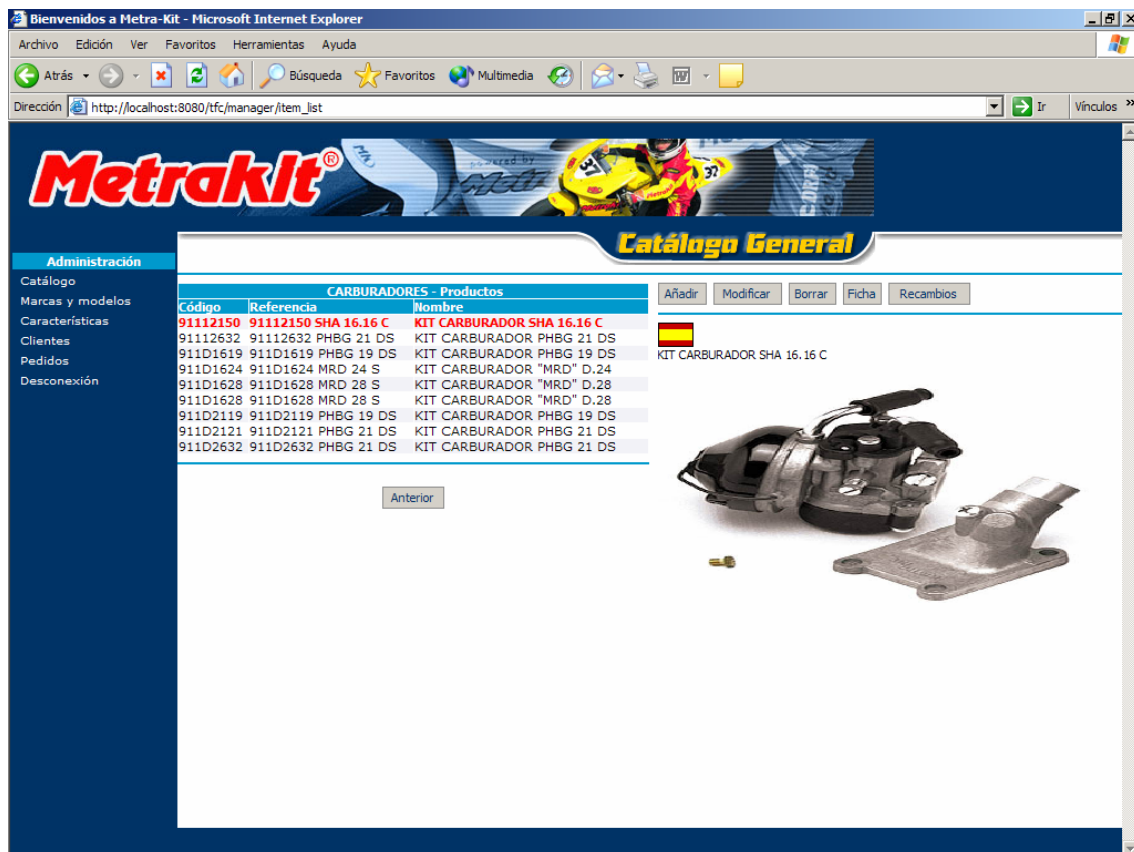
Les opcions que es poden dur a terme són

- Afegir. Per donar d'alta categories noves
- Modificar. Per modificar la categoria actual
- Esborrar. Esborra la categoria actual, prèvia confirmació de l'usuari
- Productes

Les dues primeres mostren una finestra de *detall de categoria* on es poden introduir els camps de descripcions i imatges per idiomes.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

L'opció *Productes* porta cap a la pantalla de llista de productes de la categoria en curs:



**Metrakit** *Catálogo General*

**Administración**

- Catálogo
- Marcas y modelos
- Características
- Cientes
- Pedidos
- Desconexión

Código	Referencia	Nombre
91112150	91112150	SHA 16.16 C
91112632	91112632	PHBG 21 DS
911D1619	911D1619	PHBG 19 DS
911D1624	911D1624	MRD 24 S
911D1628	911D1628	MRD 28 S
911D1628	911D1628	MRD 28 S
911D2119	911D2119	PHBG 19 DS
911D2121	911D2121	PHBG 21 DS
911D2632	911D2632	PHBG 21 DS

Añadir Modificar Borrar Ficha Recambios

KIT CARBURADOR SHA 16.16 C

Anterior




Figura 12. Relació de productes d'una categoria

La filosofia és la mateixa. Si es tria un element, aquest apareix en un altre color i es mostren les dades bàsiques d'aquest ítem a la part dreta de la pantalla (en aquest cas la descripció i la imatge). Les opcions disponibles són:

- Afegir. Per afegir un nou producte
- Modificar. Per modificar el producte en curs
- Esborrar. Per esborrar el producte en curs, prèvia confirmació de l'usuari.
- Fitxa. Manteniment de la fitxa tècnica del producte (característiques amb els seus valors)
- Recanvis. Per donar d'alta la relació de productes els quals són recanvis del producte en curs.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Les dues primeres opcions mostren una finestra de detall de producte on es poden introduir les dades bàsiques relatives al producte (codi, referència, descripcions per idioma, pes, preu):

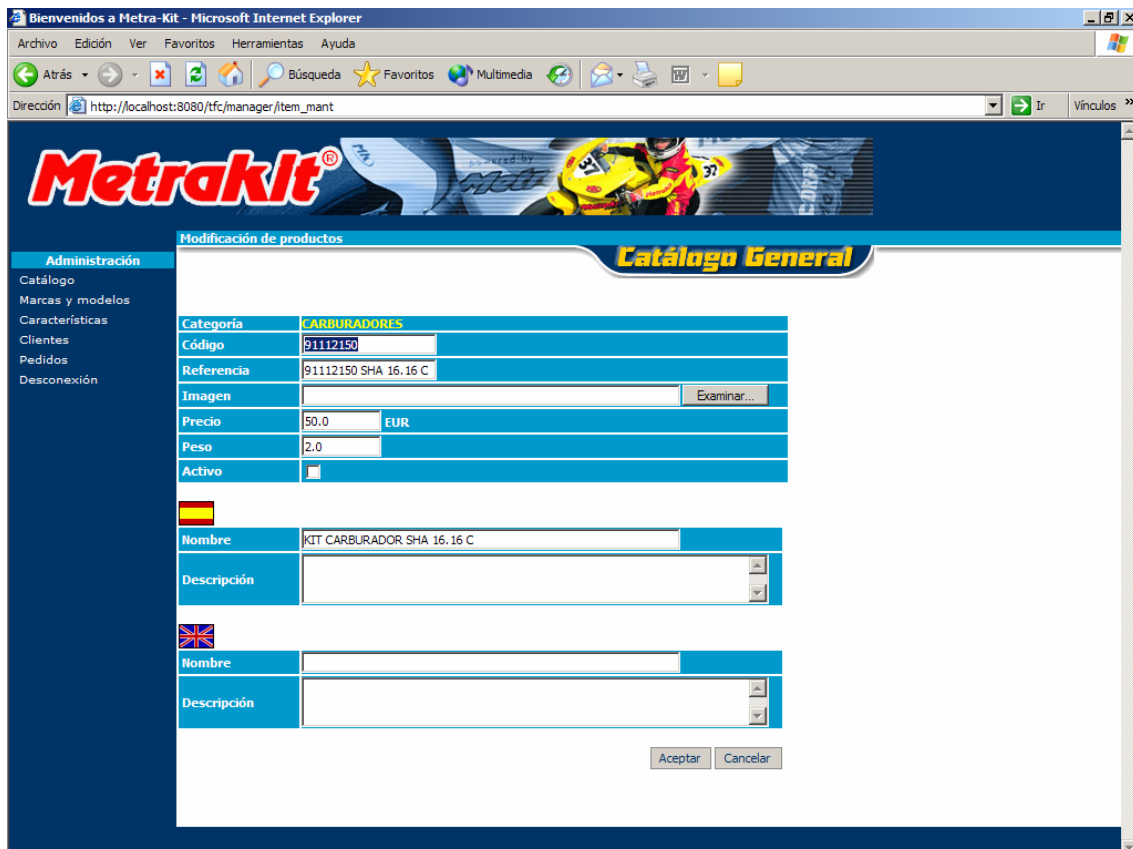


Figura 13. Pantalla de detall d'un producte

L'opció *Fitxa* permet mantenir el conjunt de característiques i els seus valors que formen la fitxa tècnica del producte:



Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

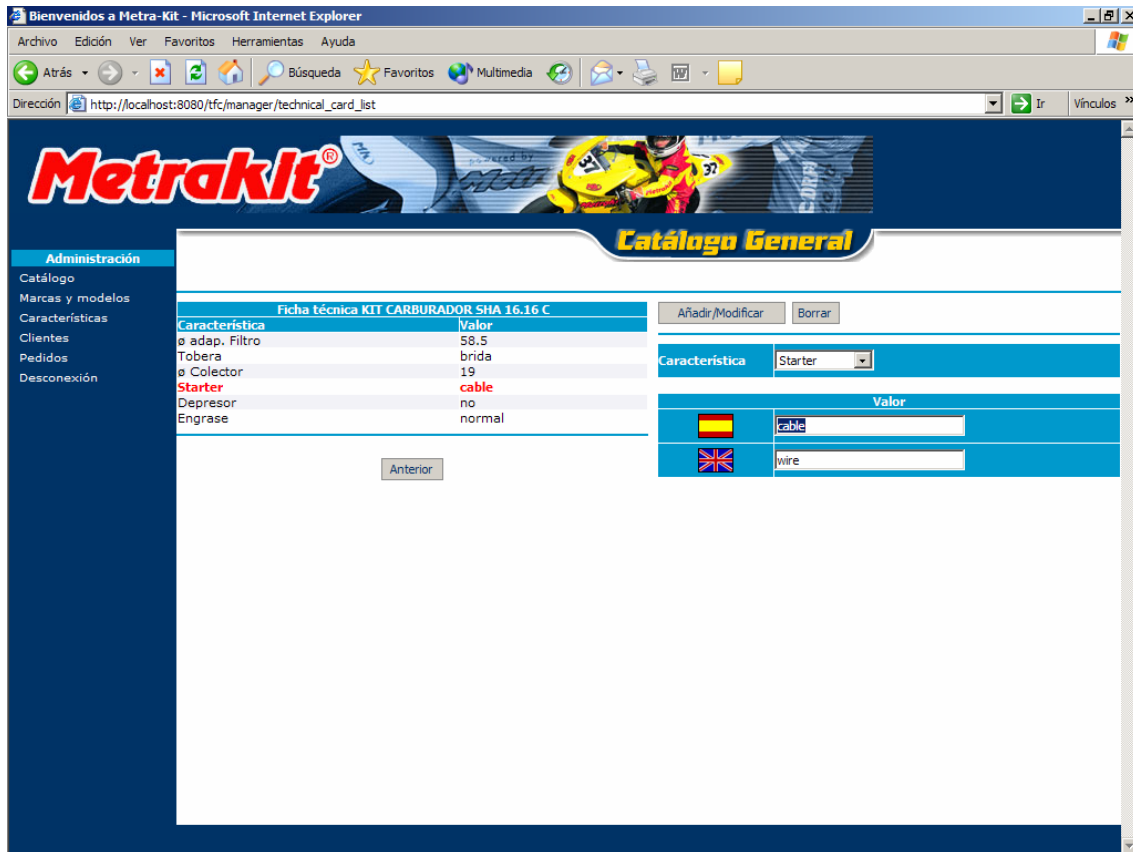


Figura 14. Manteniment de fitxa tècnica

Es pot donar d'alta una característica nova triant-la de la llista desplegable que hi ha a la part dreta de la pantalla i incorporant els valors, també depenents de l'idioma, adients. Si s'introdueix una característica ja existent, aquesta es modifica amb els nous valors. També es pot esborrar, prèvia confirmació de l'usuari, una característica concreta.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Finalment, l'opció *recanvis* presenta la següent pantalla:

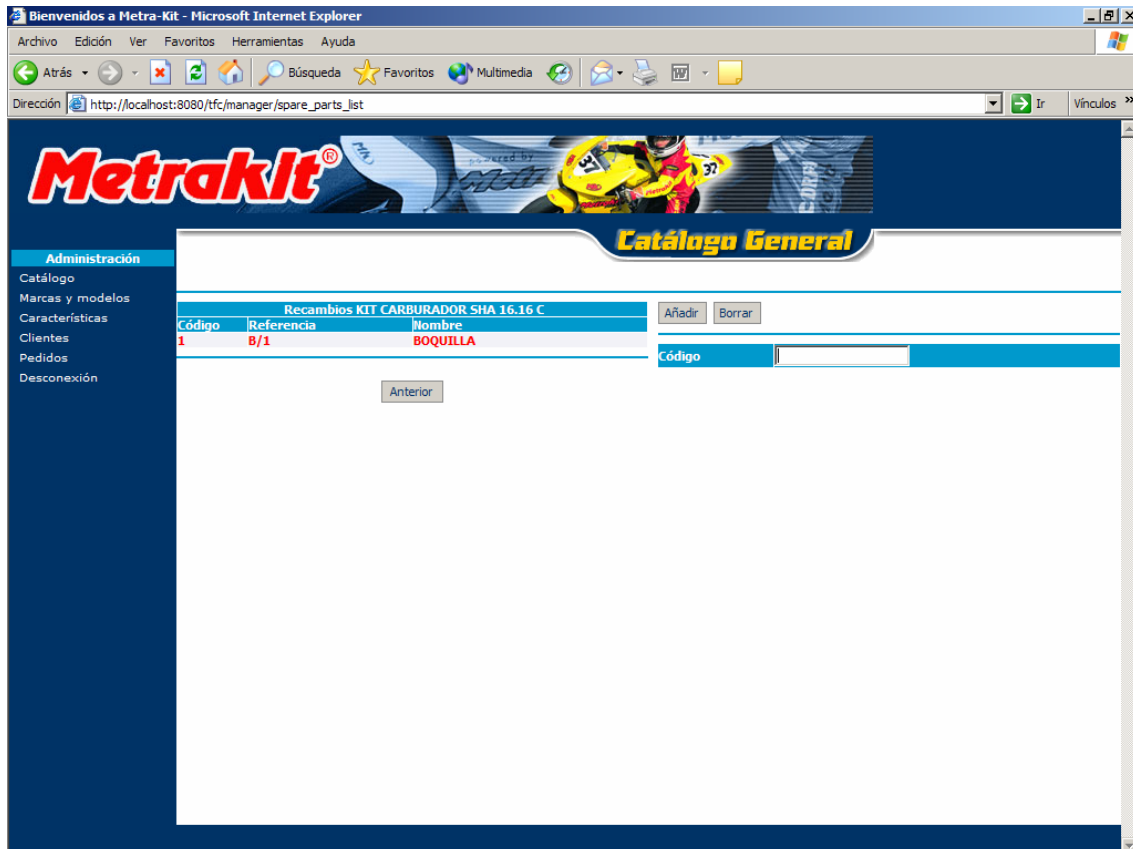


Figura 15. Manteniment de relació de recanvis d'un producte

On es poden esborrar / afegir productes recanvi del producte en curs. A l'hora d'afegir es controla que el codi ja no estigues introduït, que existeixi i que no sigui el propi el del propi producte en curs.

### 3.3.2. Marques i models

Des d'aquesta opció es poden donar d'alta les marques i models de motocicleta que poden tenir com a recanvis els productes que ofereix la botiga virtual.

Primer es mostra la relació de marques disponibles. Des d'aquí es pot modificar una marca, donar-la d'alta, donar-la de baixa i fer el manteniment dels seus models:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

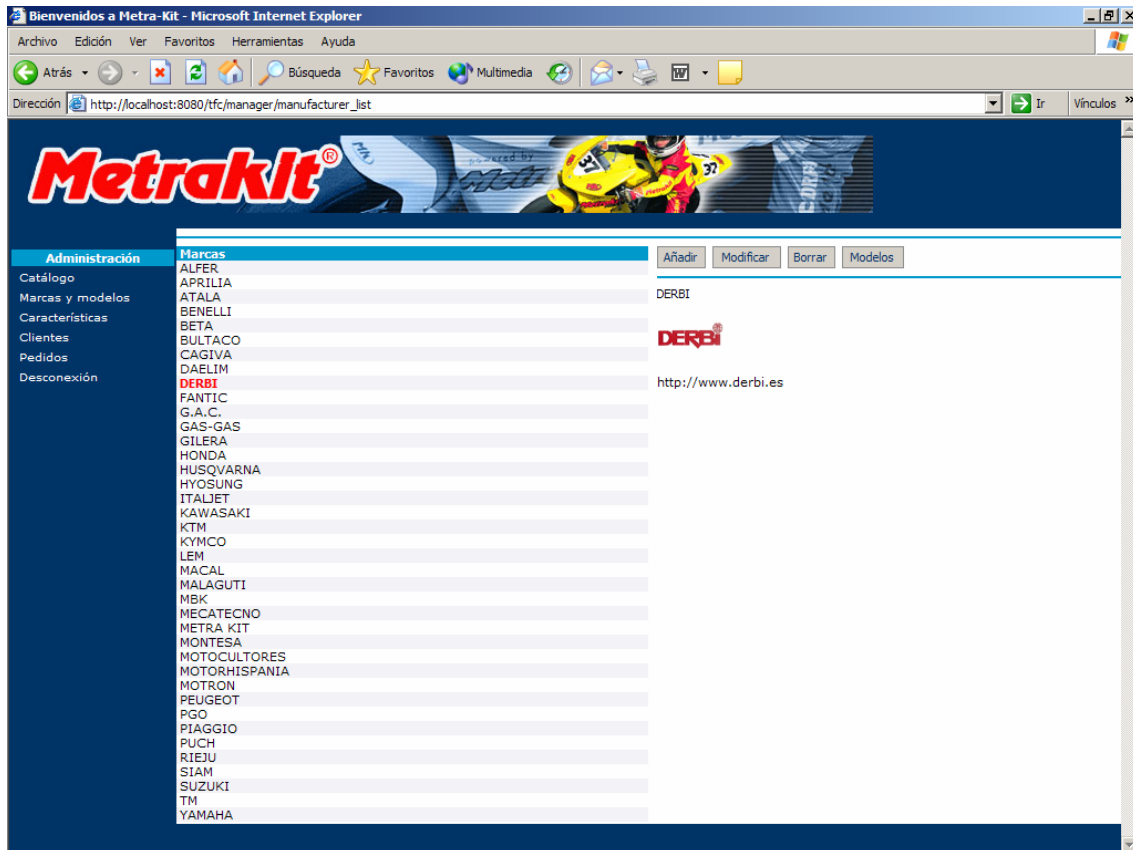


Figura 16. Relació de marques (fabricants)

En triar l'opció *afegir* o *modificar* es presenta una pantalla de detall de la marca on es pot introduir el nom, la imatge i la *URL* de la marca.

En triar l'opció *Models*, es passa a la pantalla amb la relació de models de la marca:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

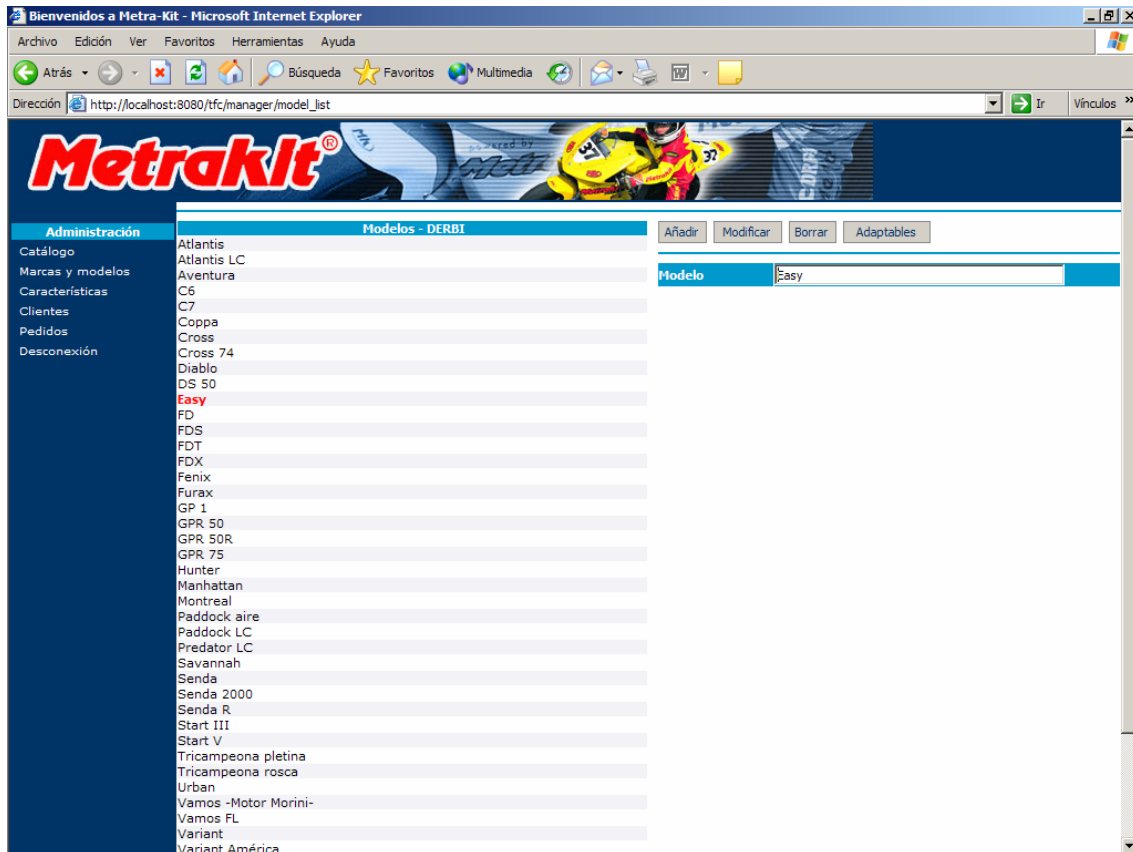


Figura 17. Relacio de models d'una marca

En aquesta pantalla es poden afegir, modificar i esborrar models. No es permet tenir 2 models amb el mateix nom.

L'opció *Adaptables* porta al manteniment de la relació de productes del catàleg que són recanvis pel model triat:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

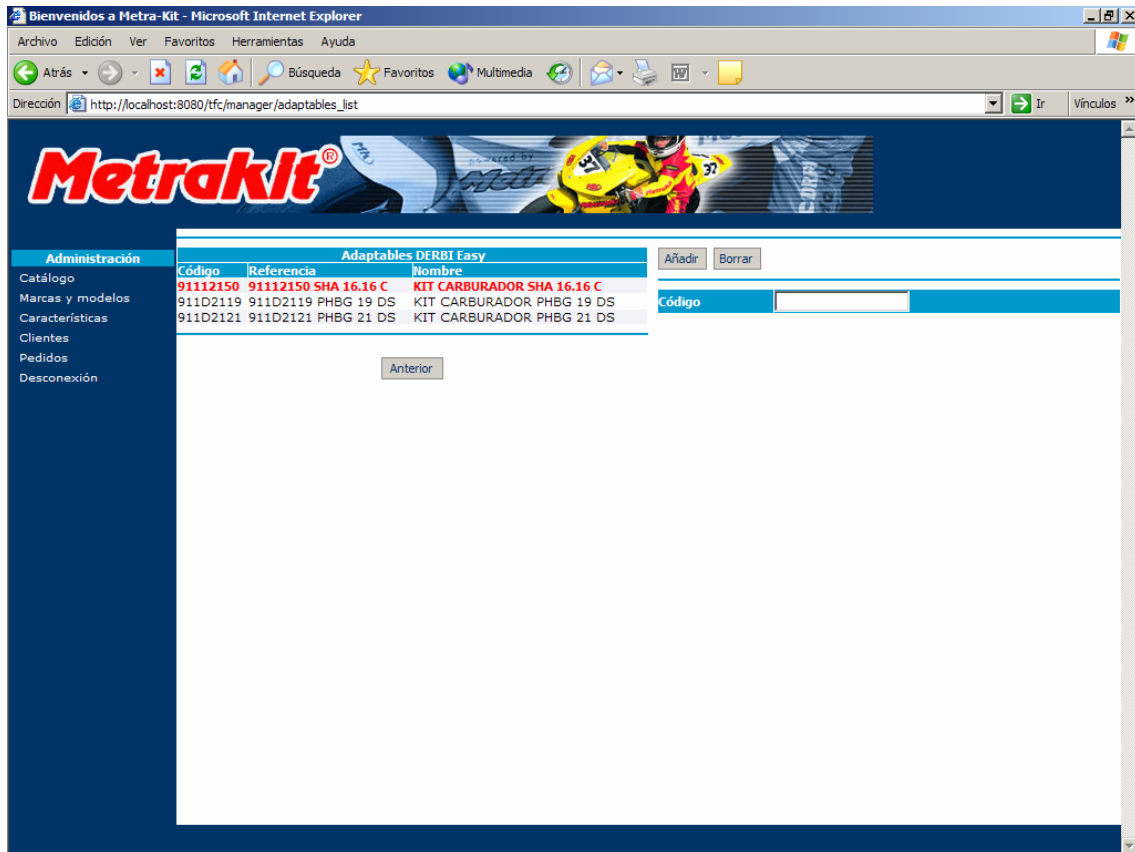


Figura 18. Relació de recanvis (adaptables) d'un model

El funcionament d'aquesta pantalla és igual a la de la de *Recanvis* del manteniment de productes.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

### 3.3.3. Característiques

Des d'aquesta opció es pot mantenir la relació de les diferents característiques que podrien tenir els diferents productes del catàleg. Aquestes seran utilitzades posteriorment per a confeccionar la *Fitxa Tècnica* dels productes.

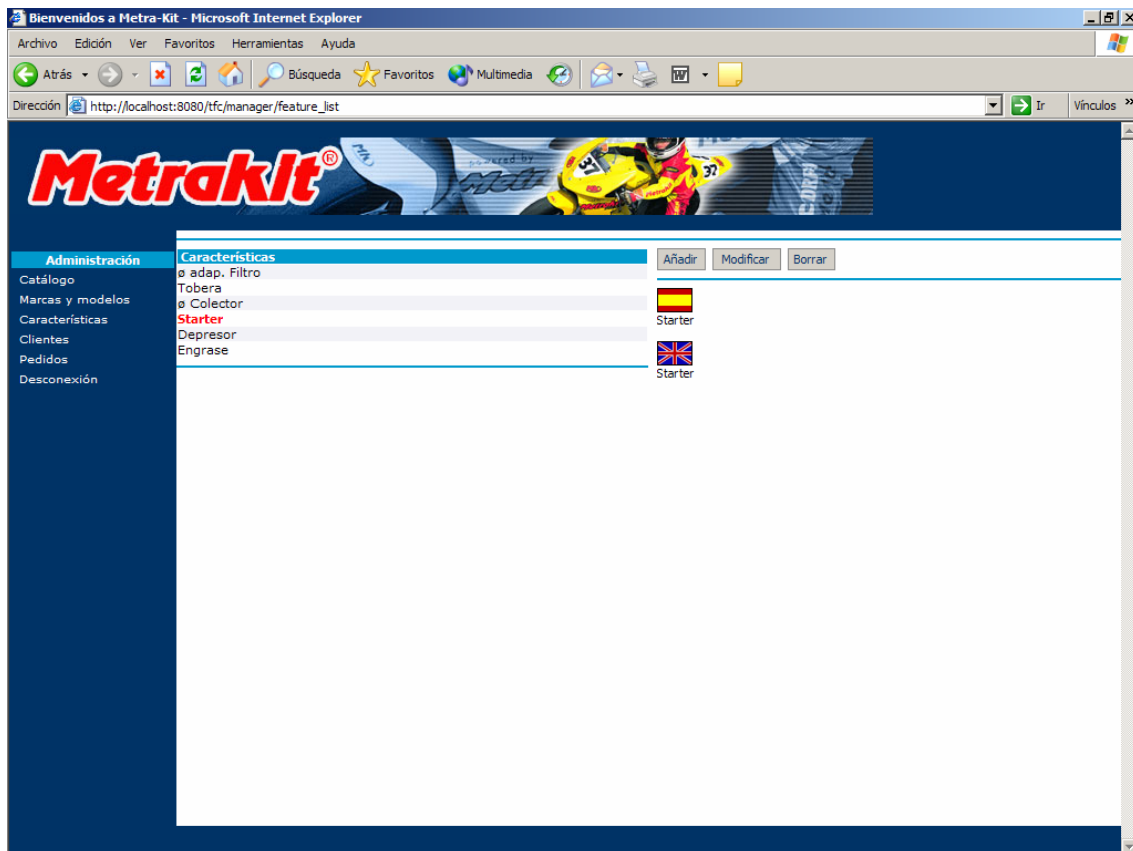


Figura 19. Manteniment de característiques

Les opcions disponibles són:

- Afegir. Per afegir noves característiques
- Modificar. Per modificar la característica actual
- Esborrar. Per esborrar la característica en curs, prèvia confirmació de l'usuari.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Les dues primeres opcions presenten una pantalla de detall de característica on es poden introduir les descripcions de les característiques en els idiomes disponibles.

### 3.3.4. Clients

Des d'aquesta pantalla es poden consultar / modificar les dades dels clients donats d'alta al sistema. Des d'aquí no es poden donar altes de client (ja que són ells mateixos que ho fan des de la *Web Pública*):

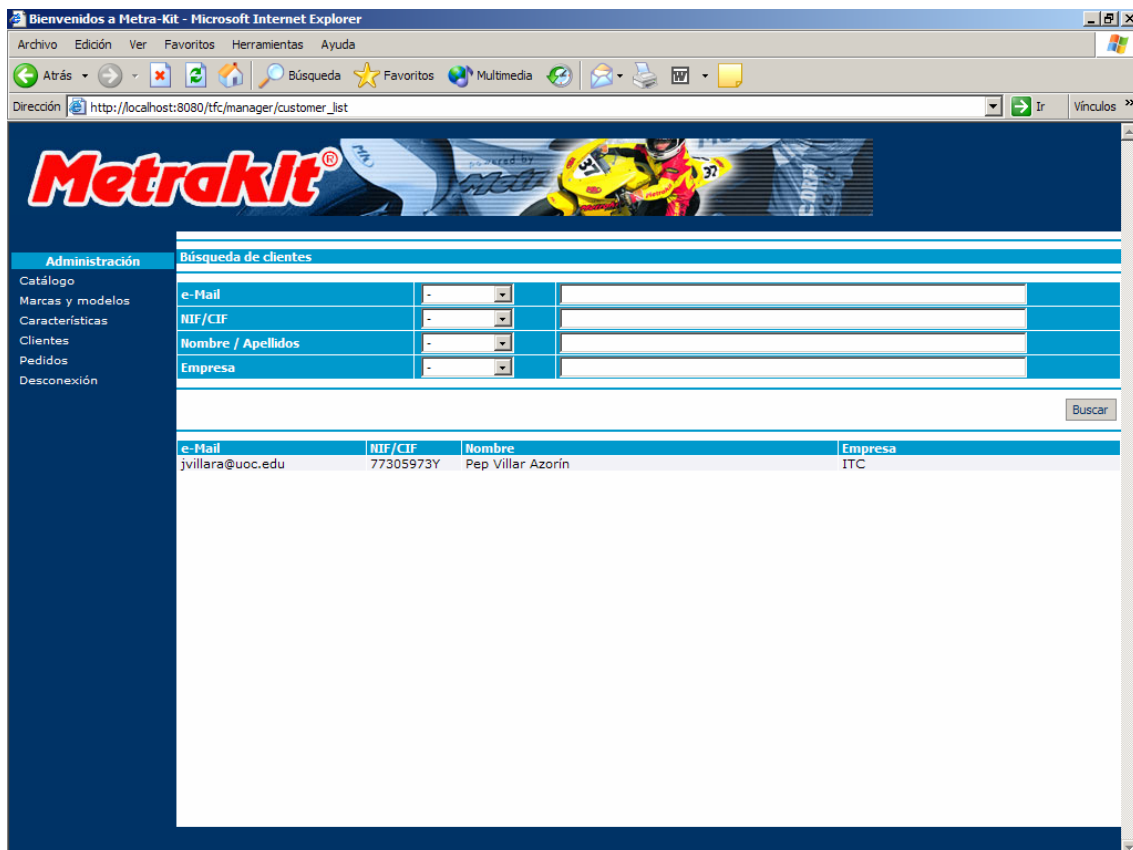


Figura 20. Clients - Cerca

Aquesta pantalla es divideix en 2 parts:

- Establiment de les condicions de cerca:

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

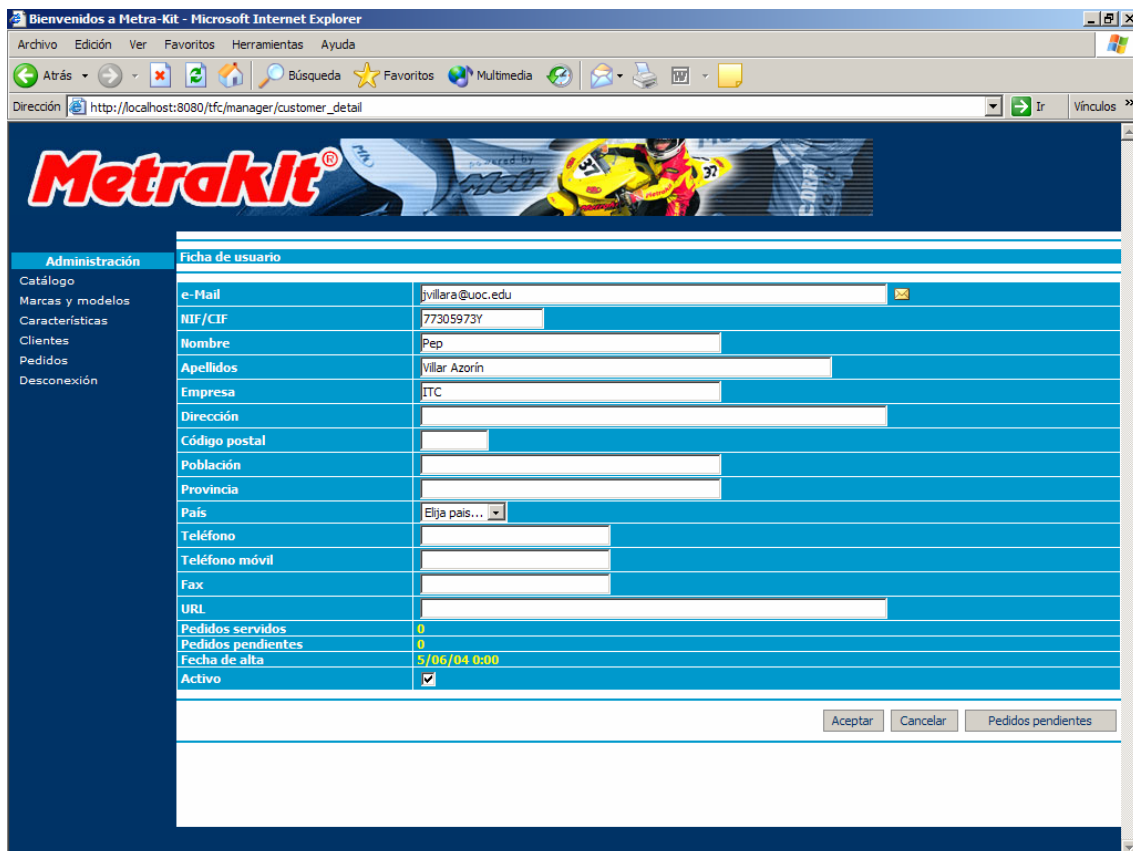
Es poden establir condicions de recerca depenent dels camps e-Mail, NIF/CIF, Nom / Cognoms, Empresa. Per a cada camp hi ha un operador que es pot triar amb una llista desplegable on els seus possibles valors són *Comença per*, *Conté* i el valor a introduir.

Es pot fer combinació de diferents camps que donarà com a resultat la conjunció (AND) de criteris.

- Llista de resultats.

La llista de resultats mostrarà els clients en funció del criteri de cerca utilitzat.

Des d'aquesta llista es pot enllaçar amb la pantalla de detall del client per consultar / modificar les seves dades:



Ficha de usuario	
e-Mail	jvillara@uoc.edu
NIF/CIF	77305973Y
Nombre	Pep
Apellidos	Villar Azorín
Empresa	ITC
Dirección	
Código postal	
Población	
Provincia	
País	Elija país...
Teléfono	
Teléfono móvil	
Fax	
URL	
Pedidos servidos	0
Pedidos pendientes	0
Fecha de alta	5/06/04 0:00
Activo	<input checked="" type="checkbox"/>

Figura 21. Clients - Detail

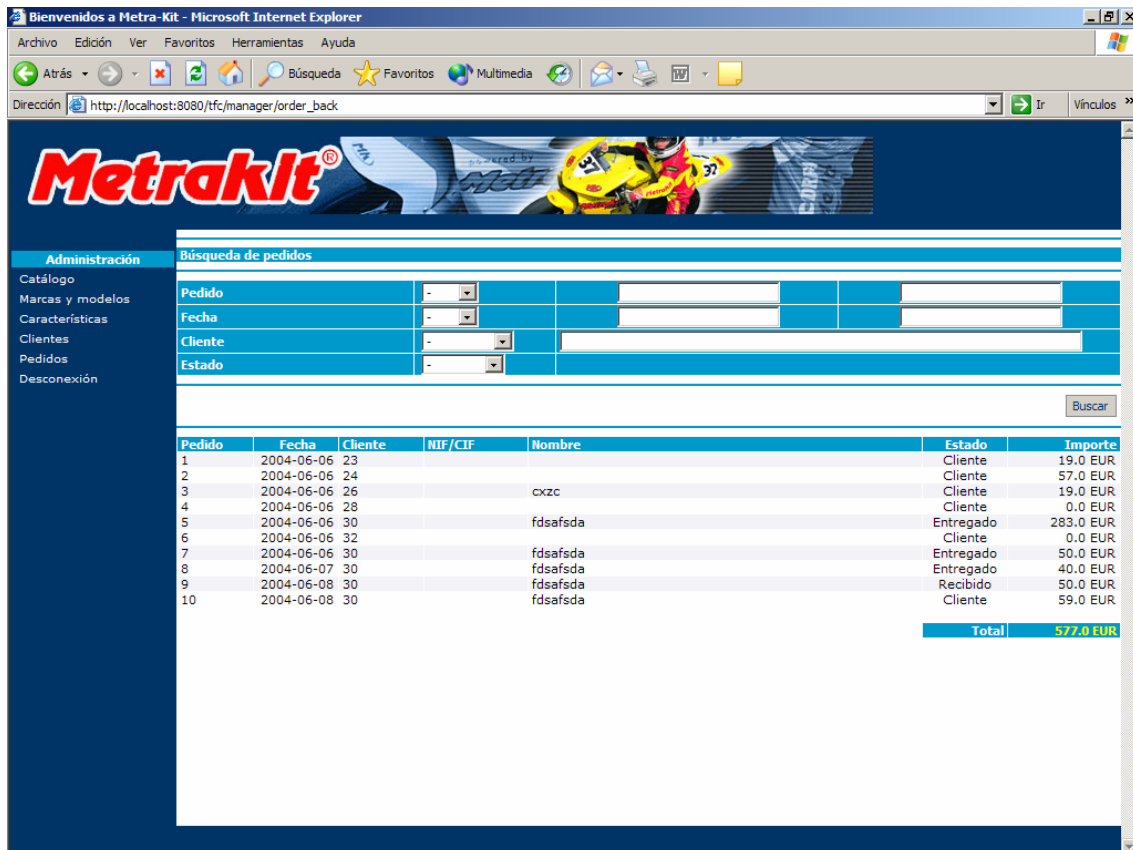


Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

Des d'aquesta pantalla també es pot consultar la relació de comandes pendents de servir de l'usuari.

### 3.3.5. Comandes

Des d'aquesta opció es poden consultar les comandes del sistema. L'usuari, a l'igual que en la pantalla de Clients, pot establir uns criteris de cerca per a facilitar la tasca de gestionar les comandes:



Pedido	Fecha	Cliente	NIF/CIF	Nombre	Estado	Importe
1	2004-06-06	23			Cliente	19.0 EUR
2	2004-06-06	24			Cliente	57.0 EUR
3	2004-06-06	26		cxzc	Cliente	19.0 EUR
4	2004-06-06	28			Cliente	0.0 EUR
5	2004-06-06	30		fdsafsda	Entregado	283.0 EUR
6	2004-06-06	32			Cliente	0.0 EUR
7	2004-06-06	30		fdsafsda	Entregado	50.0 EUR
8	2004-06-07	30		fdsafsda	Entregado	40.0 EUR
9	2004-06-08	30		fdsafsda	Recibido	50.0 EUR
10	2004-06-08	30		fdsafsda	Cliente	59.0 EUR
<b>Total</b>						<b>577.0 EUR</b>

Figura 22. Comandes - Cerca

Els criteris de cerca pels camps *Comanda* i *Data* són >, <, >=, <=, =, *entre*. Si es tria *entre* s'utilitzen els dos camps posats a tal efecte. En els altres casos només es contempla el primer camp.

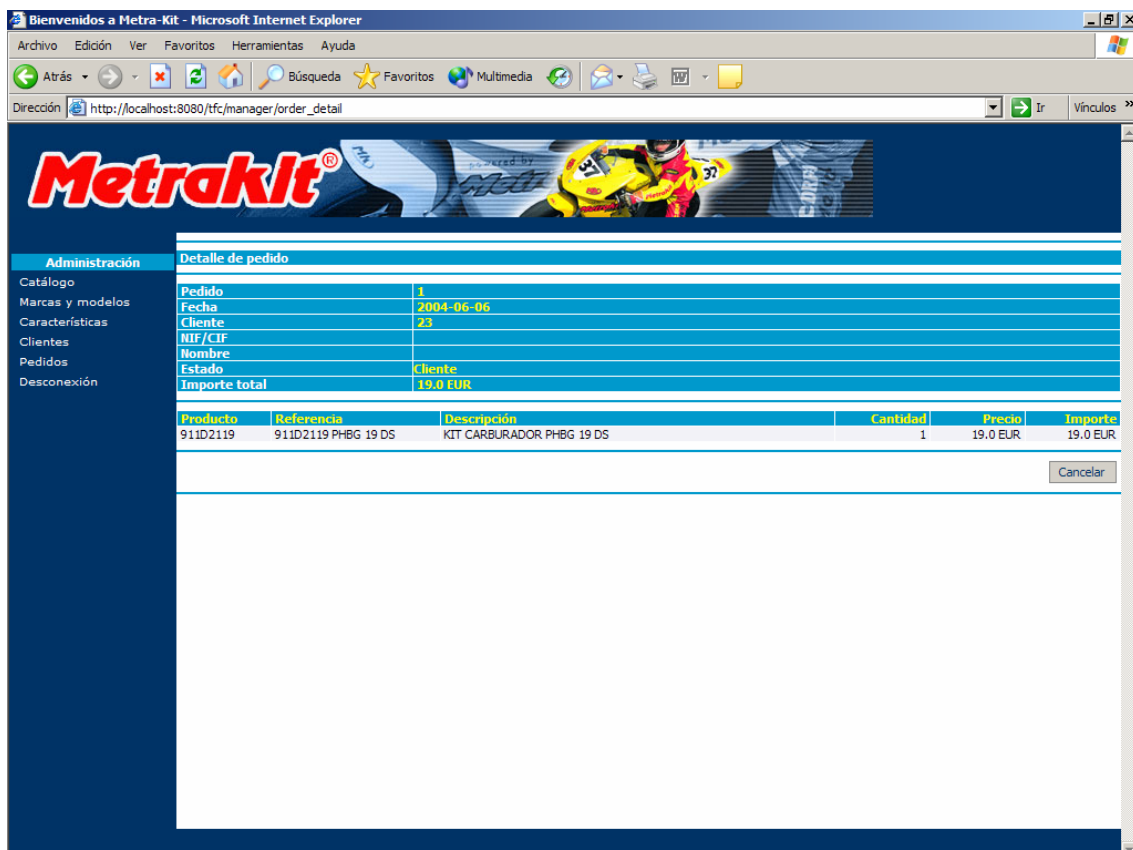
Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

El camp *Estat*, si no es posa cap, s'assumeixen tots. Si no, s'assumeix l'estat triat. Aquest criteri de cerca és útil per veure, per exemple, totes les comandes rebudes i poder-les processar (donar ordre al magatzem, etc...).

Una comanda pot tenir 3 estats:

- Client. Comanda en curs d'un client. No es pot modificar.
- Rebut. El client ha confirmat la comanda i s'està processant.
- Lliurat. S'ha processat la comanda i s'ha lliurat al client.

En entrar al detall d'una comanda, si està en estat *Client*, aquesta no es pot modificar. En cas contrari només es pot canviar l'estat.



**Detalle de pedido**

Pedido	1
Fecha	2004-06-06
Cliente	23
NIF/CIF	
Nombre	
Estado	Cliente
Importe total	19.0 EUR

Producto	Referencia	Descripción	Cantidad	Precio	Importe
911D2119	911D2119 PHBG 19 DS	KIT CARBURADOR PHBG 19 DS	1	19.0 EUR	19.0 EUR

Cancelar

Figura 23. Comandes – Detall sense modificació

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

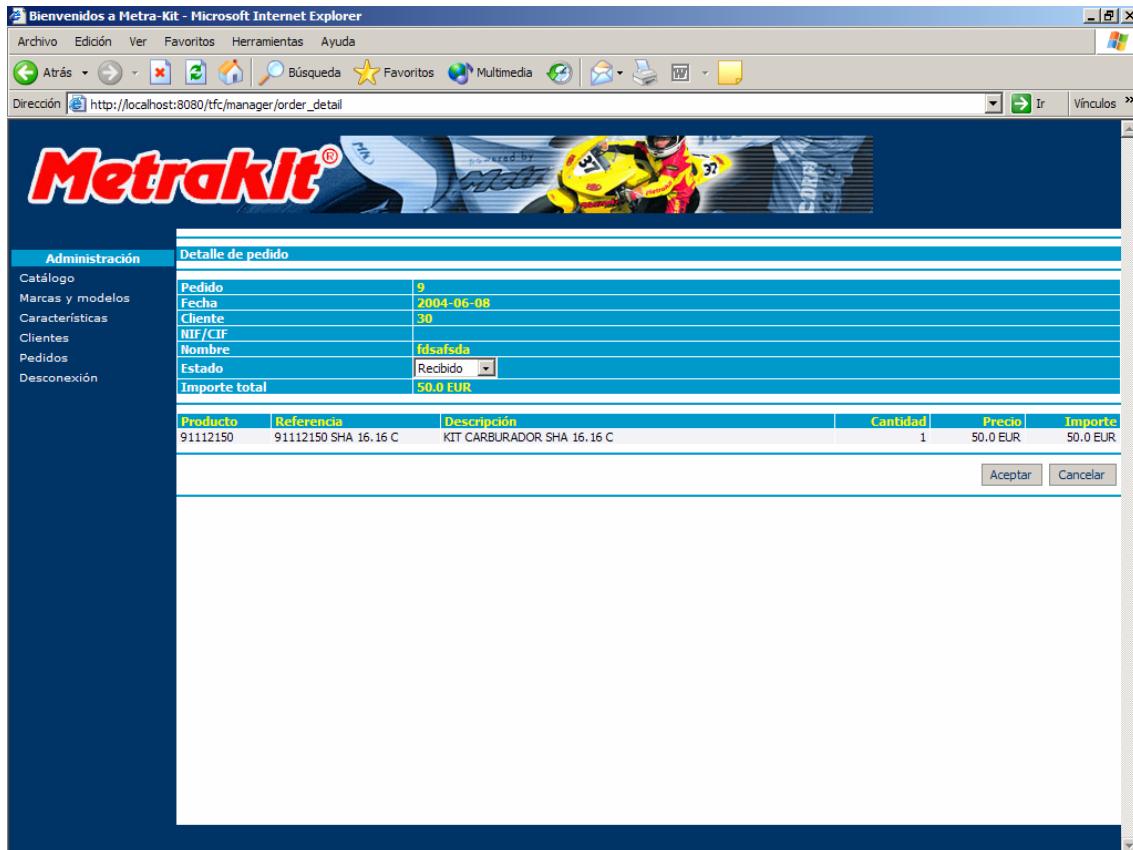


Figura 24. Comandes - Detall amb modificació

### 3.3.6. Desconnexió

Des d'aquesta opció, s'invalida la sessió actual i es torna a la pàgina principal d'administració des d'on un altre usuari s'hauria d'identificar.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## 4. Conclusions

Tot i que s'han satisfet els objectius inicials proposats en el document de definició i planificació del projecte, hi ha uns quants aspectes bàsics que s'haurien de tenir en compte per a aconseguir una infraestructura que permetés portar a terme una *botiga virtual* totalment operativa. El fet que no es cobreixin és la limitació de temps de desenvolupament del projecte.

A continuació s'enumeren aquests aspectes:

- Un major aprofitament del *framework Struts*

El *framework Struts* ofereix unes possibilitats pràcticament inabastables. En aquest projecte s'ha aprofitat aproximadament un 35% del que pot oferir. Potser la seva corba d'aprenentatge sigui gran, però els beneficis obtinguts un cop assolida encara és molt més gran.

- Gestió de traces i *logs* de l'aplicació

Un fet bàsic, especialment en una aplicació d'àmbit *públic* com pot ser una aplicació de comerç electrònic, és disposar d'un sistema de traces i *logs* d'aplicació per poder justificar tota mena de transacció econòmica que es pugui realitzar.

Aquestes traces haurien d'estar, a més, normalitzades per poder ser consultades de forma fàcil des de la part d'administració de l'aplicació.

Des del punt de vista tècnic una solució molt estesa i de cost nul és utilitzar les llibreries *log4j* que són de lliure distribució.

- Control de magatzem

Aquesta aplicació no té cap control dels *stocks* del magatzem. Si fos una aplicació de *comerç electrònic* posada a explotació, aquesta hauria de portar sense cap mena de dubte un control dels *stocks* que probablement rauria en un nou *servei*.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

- Gestió més completa dels clients

En aquest projecte s'ha proposat una gestió mínima dels clients. Seria de molta utilitat un sistema multi-adreça en el tractament de clients. Un mateix client pot tenir més d'una adreça o, per qualsevol motiu, en una comanda concreta es voldria que el lliurament fos en una adreça diferent de l'habitual.

- Gestió més completa de comandes

En cap moment en aquest projecte s'ha parlat de termes com *impostos*, *taxes*, *despeses de tramesa*, etc. S'hauria d'ampliar la gestió de les comandes (i per extensió, dels clients i dels productes) per contemplar aquests aspectes.

- Més flexibilitat en la gestió multi-idioma

Actualment l'aplicació ofereix la característica multi-idioma en funció de l'idioma del navegador que fa servir l'usuari. Només contempla castellà i anglès, i l'usuari no ho pot triar explícitament, si no depenent de la configuració del seu navegador. Seria molt més flexible si l'idioma no es delegués en el mateix *framework Struts*, si no que hi hagués una gestió específica d'idioma per usuari.

- Connexió amb mòduls de pagament electrònic

S'hauria d'habilitar la possibilitat de poder fer el pagament *on-line* de la comanda que s'ha tramès per part del client. Aquesta aplicació *suposa* un pagament contra reemborsament, però la majoria d'aplicacions accepten pagaments per targeta electrònica mitjançant la contractació d'un TPV virtual amb una Entitat Financera.

Josep Villar Azorín  
Universitat Oberta de Catalunya  
Enginyeria Tècnica d'Informàtica de Sistemes

## 5. Bibliografia

[Booch et al. 1999] Booch, Jacobson i Rumbaugh, “El Language Unificado de Modelado”, Addison-Wesley 1999

[Silberschatz et al. 1998] Silberschatz, Korth i Sudarshan, “Fundamentos de Bases de Datos” 3ª Edición, McGraw-Hill 1998

[Alur et al. ] Alur, Crupy y Malks, “Core J2EE Patterns”, Sun Microsystems

[Geary 2001] David M. Geary “Advanced Java Server Pages”, Prentice-Hall - Sun Microsystems 2001

[Stelting, Maassen 2002] Stephen Stelting i Olav Maassen “Applied Java Patterns”, Pearson Prentice-Hall – Sun Microsystems 2002

[Cavaness] Chuck Cavaness “Programming Jakarta Struts” O'Reilly