Universitat Oberta de Catalunya

Enginyeria en Informàtica

J2EE
Generació automàtica de codi

Alumne
Josep Bové Baiget
jboveb@uoc.edu

Consultor
Jordi Ceballos Villach
jceballos@uoc.edu

Curs: 2006-07
Data de lliurament: 15-01-2007

1. Resum del treball

Aquest projecte està enfocat a determinar l'estat actual de les principals eines de generació automàtica de codi que existeixen, analitzant les característiques principals de cada eina determinar-ne les funcionalitats.

Analitzar els formats XMI que generen les eines, determinant les diferencies que hi ha entre elles.

De totes les eines estudiades se'n seleccionen dues per a realitzar una aplicació i generar el codi corresponent per poder analitzar-lo

2. Índex de continguts

3. Índex de figures

3.1 Diagrames

3.2 Figures

3.3 Taules

4. Introducció

Des de no fa gaires anys s'estudia una nova manera de crear software en que els models són el centre del procés de desenvolupament. La idea principal és que els desenvolupadors escriguin models centrats en la lògica de l'aplicació i de manera automàtica es generi el codi de l'aplicació.

En l'actualitat hi han eines que responen a aquesta idea de desenvolupament. En l'estudi i aplicació d'aquestes eines es centrarà aquest projecte.

4.1. Justificació del PFC i context en el qual es desenvolupa

La idea de realitzar el PFC en la generació automàtica de codi J2EE és deguda a que la tendència del mercat actual es dirigeix en aplicacions distribuïdes per l'ús d'Internet de forma massiva.

La generació automàtica de codi està en els seus primers passos i resta molt de futur en aquest camp

El projecte es basarà en l'estudi de la situació en que estan les eines de generació automàtica de codi en el moment actual.

4.2. Objectius del PFC

L'objectiu principal del projecte és aprofundir en les eines MDA. Aquest objectiu principal s'ha desglossat en els següents objectius més concrets:

- Instal·lar i determinar les característiques principals de les eines actuals més utilitzades de generació automàtica de codi, fins i tot si proporcionen característiques de reenginyeria inversa.
- Analitzar els formats XMI que generen les eines, determinant les diferencies que hi ha entre elles.
- Realitzar una part d'un projecte utilitzant una o dues de les eines estudiades, generar el codi, i analitzar-lo.

4.3. Enfocament i mètode seguit

Per poder obtenir el màxim profit de les eines a estudiar, primer es realitzarà un estudi sobre una dotzena d'eines i després se'n triaran dues per fer l'aplicació de prova.

4.4. Planificació del projecte

La planificació del treball està dividida en 5 fases:

- **Fase 1** (Pla de treball) :

  Realització del pla de treball.

- **Fase 2** (Pac2) :

  Llistat de les principals característiques d'una eina MDA

  En aquest apartat triem l'eina MagicDraw, llistem les principals característiques en generació automàtica de codi i en reenginyeria inversa, comentant-les breument.

  Comparativa de les principals eines MDA

  En aquest apartat en triem dotze de les més representatives del mercat actual en MDA i construïm una taula amb les característiques esmentades en l'apartat anterior, d'aquesta manera podrem determinar millor les diferències que hi ha entre elles.

- **Fase 3** (Pac2) :

  Comparar el format XMI que generen

  Anàlisi del format XMI que generen les eines de l'apartat anterior, determinant les diferencies que hi ha entre elles.

- **Fase 4** (Pac3) :

  Estudi de l'aplicació a realitzar "Gestió de productes d'un catàleg"

  Realitzar l'anàlisi i disseny de l'aplicació

  Realitzar els diagrames pertinents per que l'eina MDA generi el codi que volem per l'aplicació

  Generar el codi

  Analitzar el codi generat

  S'analitza quin codi genera i quin no i del que genera, comprovar que ho realitza correctament.

- **Fase 5** (Entrega Final) :

Recopilació de la documentació i realització de la memòria i la presentació virtual.

En el diagrama de Gantt següent es poden veure de manera gràfica les diferents fases del projecte:

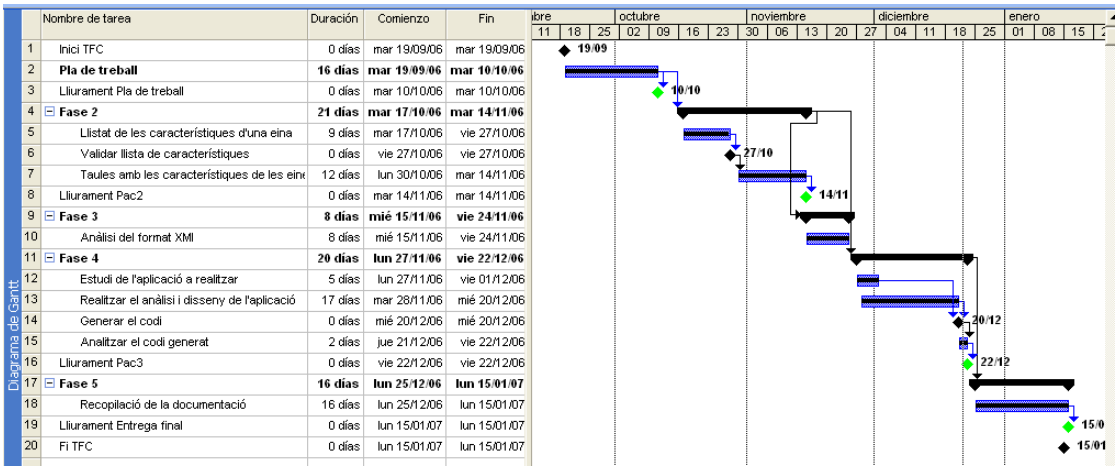


Diagrama 1: Diagrama de Gantt

4.5. Productes obtinguts

Els fitxers que s'han generat com a resultat d'aquest treball són següents:

jboveb_memoria.zip fitxer on hi ha la memòria

jboveb_presentacio.zip fitxer on hi trobem la presentació amb diapositives

jboveb_producte.zip fitxer on hi trobem el producte resultat de l'estudi i està compost per dos fitxers que diferencien l'eina utilitzada.

jboveb_producte_ArgoUml.zip fitxer amb el producte que s'ha realitzat amb l'eina ArgoUML.

jboveb_producte_MagicDraw.zip fitxer amb el producte que s'ha realitzat amb l'eina MagicDraw

Estructura de directoris dels productes:

En el directori *jboveb_producte_ArgoUML* hi tenim el fitxers que contenen els diagrames que necessita ArgoUML per generar el codi que volem.

En el directori *jboveb_producte_MagicDraw* hi han el fitxers que contenen els diagrames que necessita MagicDraw per generar el codi que volem.

En el directori *sql* hi ha el fitxer generat per MagicDraw de la construcció de les taules de la base de dades.

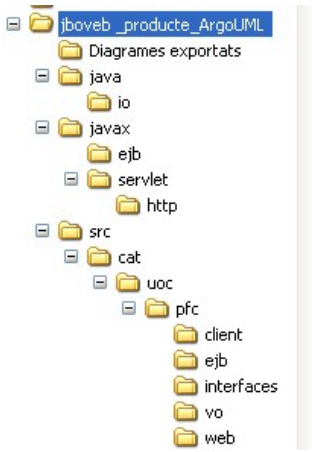
En el directori *icons* hi han els icons que utilitza MagicDraw per realitzar els diagrames.

En el directori *src* i els seus subdirectoris hi han el fitxers java que han generat les eines.

En el directori *WEB-INF* hi han el fitxers descriptors que necessita J2EE per llençar l'aplicació i que ha generat MagicDraw.

En el directori *java*, *javax* i els seus subdirectoris hi han el fitxers java que ha generat ArgoUML i que pertanyen als paquets pertinents.

jboveb_Producte_ArgoUml.zip



jboveb_Producte_MagicDraw.zip

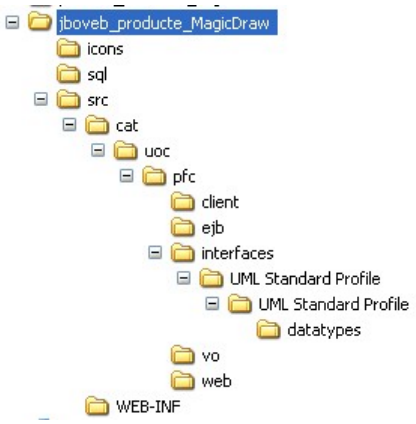


Figura 1: Arbres del directoris dels productes

4.6. Breu descripció dels altres capítols de la memòria

En els propers dos capítols (5 i 6) es descriu el cos del projecte i en els següents (7 i 8) les conclusions obtingudes.

En el capítol 5 "*Descripció de les eines*" es realitza l'estudi de les característiques principals de generació automàtica de codi d'una dotzena d'eines i un resum del format XMI que utilitza cada una d'elles.

En el capítol 6 *"Realització d'una aplicació"* es realitza l'anàlisi i disseny d'una aplicació de gestió de productes d'un catàleg i s'utilitzen dues eines per la generació del codi de l'aplicació. Les eines triades per realitzar les proves han estat: MagicDraw i ArgoUML.

En el capítol 7 *"Conclusions"* s'exposen les conclusions extretes de l'estudi de les característiques de les eines i l'aplicació de les eines per la generació de codi.

En el capítol 8 *"Línies futures de treballs"* s'exposa quina direcció seguiran les eines MDA en el futur.

5. Descripció de les eines

A continuació es donarà una petita descripció de totes les eines que s'estudiaran en aquest projecte.

Eina	Link
Altova Umodel	http://www.altova.com/features_reverse_engineer.html
ArgoUML	http://argouml.tigris.org
JUDE	http://jude.change-vision.com/jude-web/index.html
MagicDraw	http://www.magicdraw.com
Metamill	http://www.metamill.com
NetBeans Enterprise	http://www.netbeans.org/kb/55/uml_re.html
Objecteering	http://www.objecteering.com/objecteering6.php
Poseidon	http://www.gentleware.com
Rational Rose	http://www.rational.com
StarUML	http://staruml.sourceforge.net
Sun Java Studio Enterprise	http://developers.sun.com/prodtech/javatools/jsenterprise/index.jsp
Visual Paradigm	http://home.agh.edu.pl/~olekb/vpapp/

Taula 1: Adreces URL de les eines estudiades

➢ Altova Umodel 2006 rel. 2

Eina de desenvolupament d'aplicacions i software de modelatge UML 2.1.

Dissenya visualment models d'aplicacions i genera codi. Inclou reenginyeria inversa amb capacitat de llegir codi font i generar models UML.

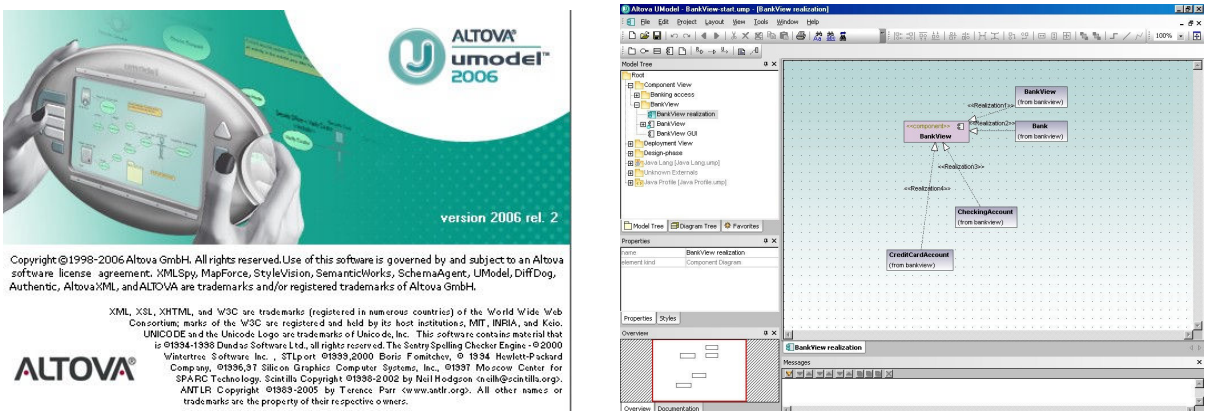


Figura 2: Logotip i una pantalla d'Altova Umodel

http://www.altova.com/features_reverse_engineer.html

➢ ArgoUML V0.22

Eina professional de modelatge UML, escrit en Java i dissenyat amb els criteris de codi obert i llicencia lliure.
Dissenya visualment models d'aplicacions i genera codi. Inclou reenginyeria inversa amb capacitat de llegir codi font i generar models UML.

S'executa directament des de Java "c:\ArgoUml\java –jar argouml.jar"

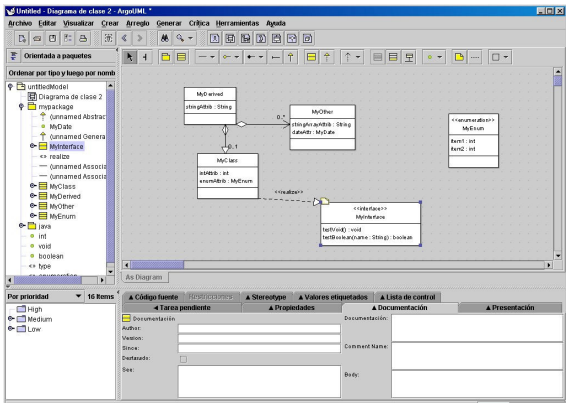


Figura 3: Logotip i una pantalla d'ArgoUML

http://argouml.tigris.org

➢ JUDE Professional 3.1b2

Eina de modelatge UML, d'ús amigable construïda amb el concepte d'alta usabilitat des de la instal·lació. Suporta software de disseny orientat a objectes en Java combinat amb Mind Map.

Dissenya visualment models d'aplicacions i genera codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.

Hi ha disponibles diferents versions: Professional, Server i Community

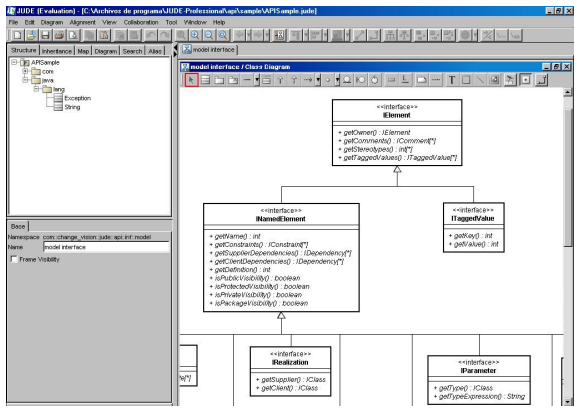


Figura 4: Logotip i una pantalla de JUDE

http://jude.change-vision.com/jude-web/index.html

➢ MagicDraw UML Enterprise 11.6

Eina visual de modelatge UML i eina CASE amb suport per treball en equip, facilita l'anàlisi i el disseny orientat a objectes i sistemes de bases de dades. Dissenya visualment models d'aplicacions i genera codi, inclou reenginyeria inversa amb capacitat de generar models UML a partir de codi font.

Hi ha disponibles diferents versions: Reader, Personal, Standard, Professional, Community i Enterprise

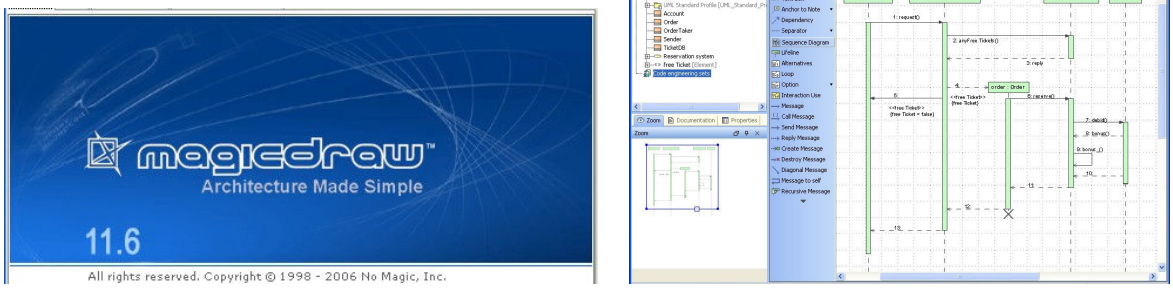


Figura 5: Logotip i una pantalla de MagicDraw

http://www.magicdraw.com

➢ Metamill UML CASE Tool V4.2

Eina de modelatge UML. Captura els requeriments del negoci utilitzant diagrames estàtics i dinàmics

Eina professional de baix cost. Suporta la creació de models visuals

Dissenya visualment models d'aplicacions i genera codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.

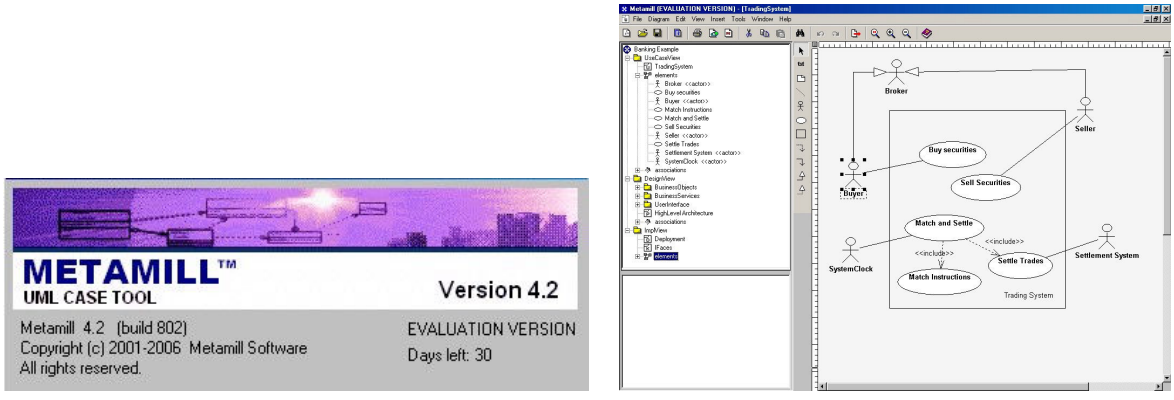


Figura 6: Logotip i una pantalla de Metamill

http://www.metamill.com

➢ NetBeans Enterprise Pack 5.5 + NetBeans IDE 5.5

NetBeans Enterprise és un pack que s'integra al NetBeans IDE i conjuntament són una eina d'escriptura, prova i depuració d'aplicacions SOA utilitzant XML, BPEL i serveis web java, amb capacitat de modelatge UML.

Dissenya visualment models d'aplicacions i genera codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.

Hi ha disponibles diferents versions: Reader, Personal, Standard, Professional, Community i Enterprise

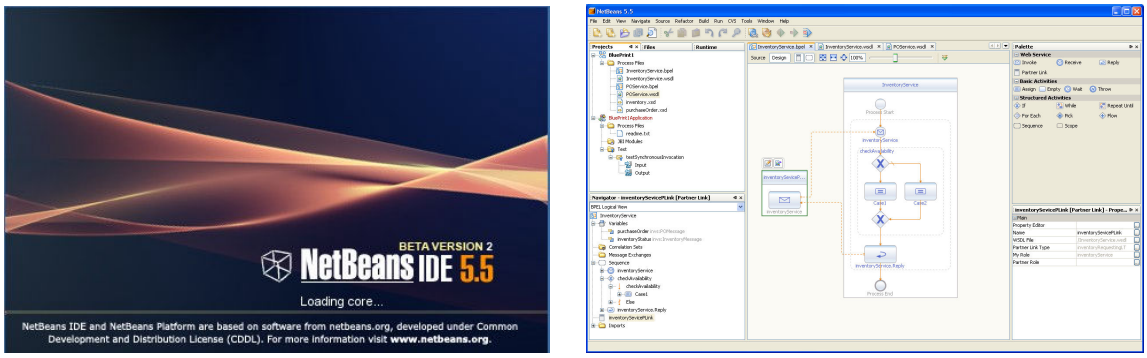


Figura 7: Logotip i una pantalla de NetBeans

http://www.netbeans.org/kb/55/uml_re.html

➢ Ogjecteering 6 Enterprise Edition

Eina MDA de desenvolupament i modelatge UML. Proporciona una màxima cobertura de MDD incorporant les ultimes versions d'UML, MDA, MDD, és integrable amb eines de desenvolupament IDE.

Controla la consistència del model, la traçabilitat en totes les parts del cicle de vida del programari (requeriments, anàlisi, generació de codi i proves) en el desenvolupament de l'aplicació. Proporciona nombroses solucions i permet crear les seves pròpies extensions UML per adaptar-les millor al negoci. Dissenya visualment models d'aplicacions i genera codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.

Hi ha disponibles diferents versions: Enterprise i Free Edition.

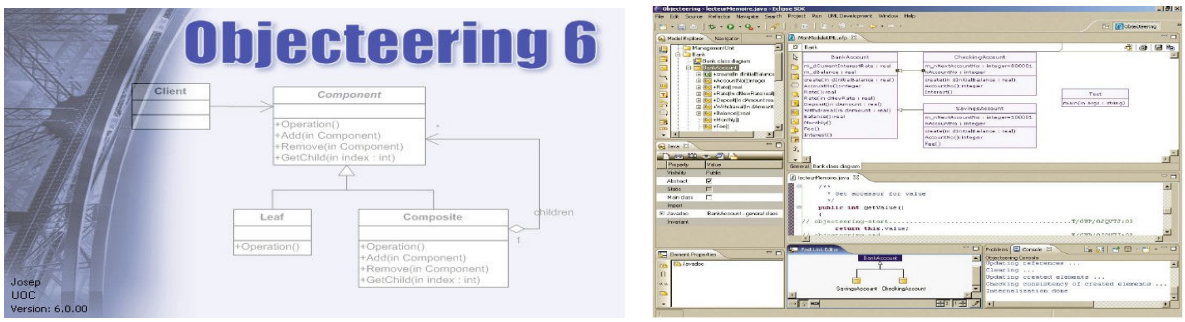


Figura 8: Logotip i una pantalla d'Ogjecteering

http://www.objecteering.com/objecteering6.php

➢ Poseidon Professional 4.2

Eina de modelatge UML per l'anàlisi, disseny i documentació del procés de desenvolupament d'una aplicació que proporciona una alta productivitat, i una interface molt intuïtiva. Dissenya visualment models d'aplicacions i genera codi, inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.

Hi ha disponibles diferents versions: Community, Standard, Professional, Enterprise i Embedded Enterprise

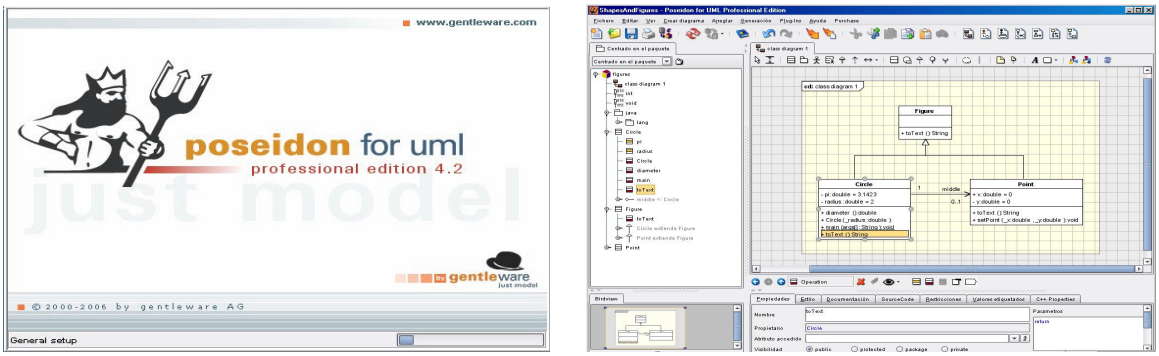


Figura 9: Logotip i una pantalla de Poseidon

http://www.gentleware.com

➢ Rational Rose Enterprise Edition

És una eina de desenvolupament basada en models que s'integra a base de dades i IDEs de les principals plataformes. Ofereix un llenguatge de modelat propi per agilitzar la creació d'aplicacions.

Dissenya visualment models d'aplicacions i genera codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.

Hi ha disponibles diferents versions: Modeler, Developer i Enterprise.

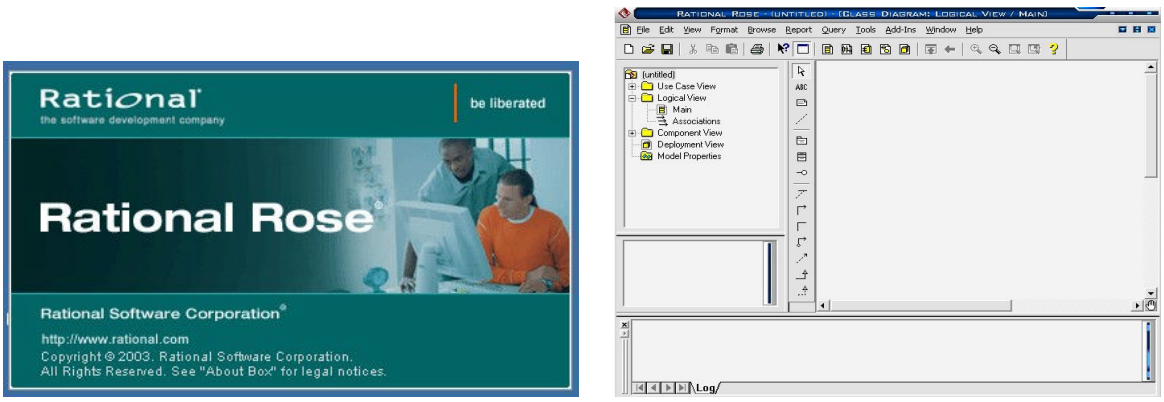


Figura 10: Logotip i una pantalla de Rational Rose

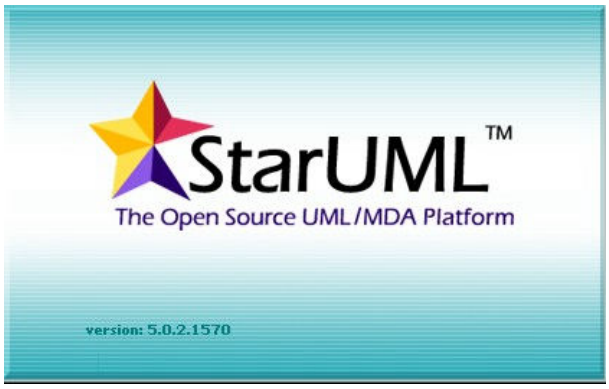
http://www.rational.com

➢ StartUML 5.0

És una eina MDA de desenvolupament i modelatge UML. Proporciona un desenvolupament ràpid, flexible, extensible i acurat que garanteix la màxima productivitat i qualitat dels projectes

StratUML és una eina que s'adapta a l'usuari, oferint-li unes variables configurables de la interface a utilitzar.

Dissenya visualment models d'aplicacions i genera el codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.



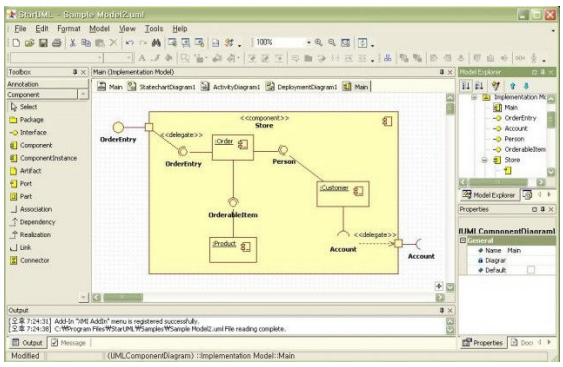


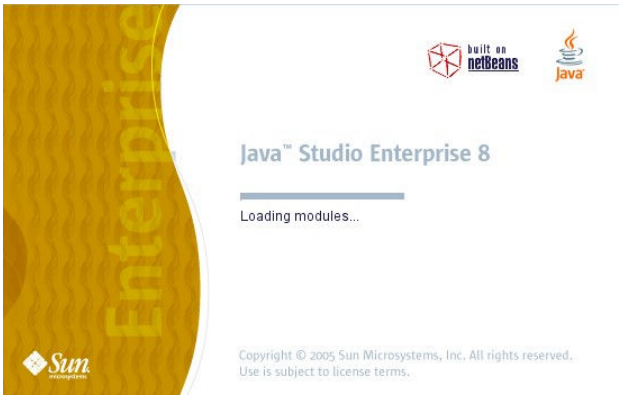
Figura 11: Logotip i una pantalla de StartUML

http://staruml.sourceforge.net

➢ Sun Java Studio Enterprise 8

Sun Java Studio Enterprise integrada amb un IDE constitueix una eina de desenvolupament i modelatge UML, ofereix les funcionalitats necessàries per desenvolupar, depurar, provar i realitzar el seguiment de les aplicacions. Integra AVK.

Dissenya visualment models d'aplicacions i genera el codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.



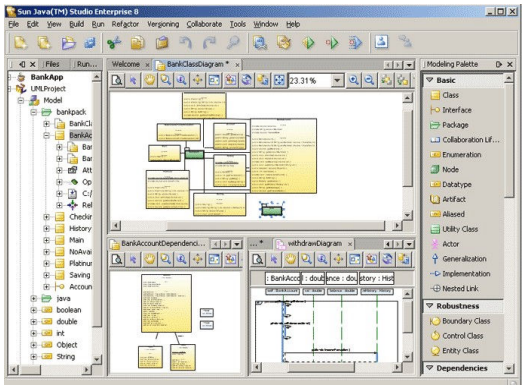


Figura 12: Logotip i una pantalla de Java Studio

http://developers.sun.com/prodtech/javatools/jsenterprise/index.jsp

➢ Visual Paradigm Enterprise 5.3

Eina CASE de desenvolupament d'aplicacions i software de modelatge UML Té unes excel·lents possibilitats d'intercanvi amb altres eines CASE. Es fàcil d'utilitzar i de ràpid aprenentatge.

Dissenya visualment models d'aplicacions i genera el codi. Inclou reenginyeria inversa amb capacitat de generar models UML a partir del codi font.

Hi ha disponibles diferents versions, Professional, Standard, Modeler, Personal, Community

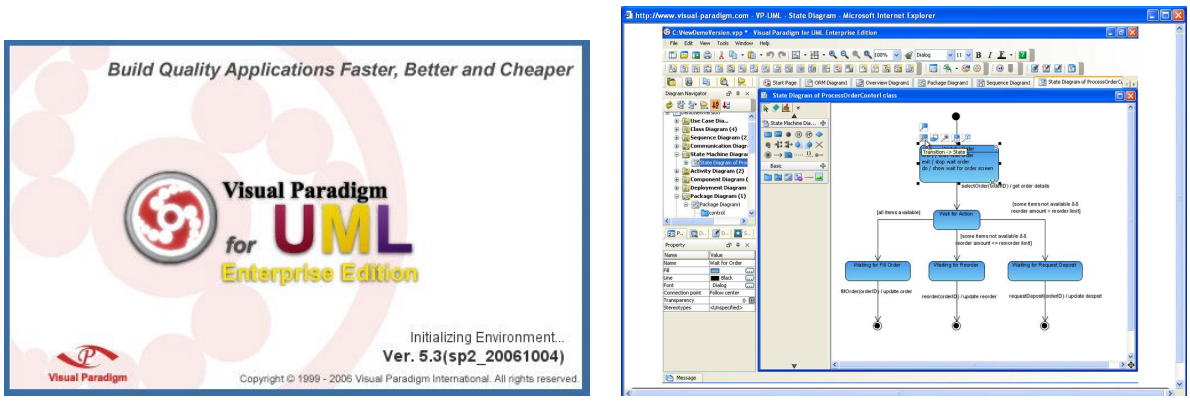


Figura 13: Logotip i una pantalla de Visual Paradigm

http://home.agh.edu.pl/~olekb/vpapp/

5.1. Característiques de l'estudi

- **Generals**
  - **URL**
    Direcció URL de l'empresa
  - **Preu**
    Preu venda al públic de l'eina
  - **Patrons**
    Tipus de patrons (patterns) que pot utilitzar l'eina
  - **Plataformes**
    S'especifica en quines plataformes pot ser instal·lada l'eina
  - **Extensibilitat**
    S'especifica per quin mitjà es poden fer unes extensions per obtenir més funcionalitats de les que presenta l'eina en la versió original.

- **Generació de codi**
  - **XMI**
    Versió de l'especificació XML d'intercanvi de metadades que exporta i importa l'eina
  - **UML**
    Versió de l'especificació del llenguatge unificat de modelatge que exporta i importa l'eina
  - **Estereotips**
    Especifica si suporta estereotips
  - **Restriccions OCL**
    Especifica si suporta restriccions (constraints), pre-condicions, post-condicions i invariants
  - **Diagrames UML**
    Indica els tipus de diagrames UML a partir dels quals és capaç de generar codi
  - **Altres tipus de diagrames**
    Indica altres tipus de diagrames a partir dels quals és capaç de generar codi
  - **Llenguatges**
    Indica el llenguatge que pot utilitzar a l'hora de generar el codi automàticament
  - **Framework**
    Indica si suporta marcs o esquelets estàndards del mercat
  - **Perfils UML**
    Indica si suporta perfils (profiles)
  - **Tagged values**
    Valors etiquetats. Element estàndard de l'especificació d'UML metamodel
- **Reenginyeria inversa**
  - **Llenguatges**
    Indica el llenguatge que pot utilitzar a l'hora de realitzar una reenginyeria inversa
  - **Diagrames**
    Especifica els tipus de diagrames que pot generar des del codi generat amb una reenginyeria inversa

5.2. Taules resum

Característiques	Altova Umodel 2006 r0.22
Generals	
URL	http://www.altova.com
Preu	€123.75
Patrons	No
Plataformes	Windows
Extensibilitat	No
Generació de codi	
XMI	Suporta el XMI 2.1, importa i exporta XMI 1.2
UML	Suporta UML 2.1, és compatible amb UML 2.0
Estereotips	Si
Restriccions OCL	Si
Diagrames UML	Seqüència, Classes, Casos d'ús, Components, Objectes, Estats, Implementació (components i desplegament)
Altres tipus de diagrames	Paquets, Comportament, Arquitectura, Estructura de components
Llenguatges	Java, C#
Framework	No
Perfils UML	Si
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java, C#
Diagrames	Genera diagrames per paquets, estructura de dades, relacions Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 2: Característiques de l'eina Altova Umodel

Característiques	**ArgoUML 0.22**
Generals	
URL	http://argouml.tigris.org
Preu	Open-source (llicència BSD "*MIT Licence*") (Eina escrita amb Java)
Patrons	GoF
Plataformes	Plataforma virtual amb java 1.4 o java 5.0
Extensibilitat	API
Generació de codi	
XMI	Importa l'estàndard XMI 1.2 i exporta l'estàndard XMI 1.2
UML	Està basat en l'especificació UML 1.4
Estereotips	Si
Restriccions OCL	Si (pre, post, invariant)
Diagrames UML	Classes, Casos d'ús, Estats, Activitat, Paquets, Components, Objectes, Col·laboració, Seqüència
Altres tipus de diagrames	No
Llenguatges	Java, C++, C#, PHP4, PHP5
Framework	Si (utilitzat per la implementació)
Perfils UML	Si
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java (.class, .jar)
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 3: Característiques de l'eina ArgoUML

Característiques	JUDE Professional 3.1b2
Generals	
URL	http://jude.change-vision.com
Preu	$280
Patrons	GoF
Plataformes	Windows (32 bits)
Extensibilitat	API
Generació de codi	
XMI	Suporta, importa i exporta un format original XMI 1.1 amb un model UML 1.4
UML	Suporta parcialment UML 2.0 i completament UML 1.4
Estereotips	Si
Restriccions OCL	No
Diagrames UML	Classes. Seqüència, Activitat, Casos d'ús, Implementació (components i desplegament), Col·laboració, Estats
Altres tipus de diagrames	Eriksson-Penker Procés, Robustesa, Mind, Estructura de components
Llenguatges	Java (java Skeleton)
Framework	No
Perfils UML	No
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java, (java font per crear models)
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 4: Característiques de l'eina JUDE

Característiques	MagicDraw UML Enterprise 11.6
Generals	
URL	http://www.magicdraw.com
Preu	€1355
Patrons	GoF = Adapter, Bridge, Composite, Decorator, Observer, proxy, Singleton, Visitor Java = Main, EJB, RMI, WSDL, CORBA, IDL Junit = Test Case, Tested Class
Plataformes	Windows, Mac, Unix
Extensibilitat	OpenAPI, Plug-in
Generació de codi	
XMI	Suporta XMI 2.1, Importa de XMI 1.2
UML	Suporta UML 2.0 metamodel i anotació, Importa de UML 1.4 metamodel
Estereotips	Si
Restriccions OCL	Si (pre, post, invariants)
Diagrames UML	Classes (paquets i objectes), Casos d'ús, Seqüències, Estats, Activitat, Implementació (components i desplegament), Comunicació
Altres tipus de diagrames	Robustesa, WSDL, genèric DDL, Oracle DDL, XML Schema, Web application, Time, Corba, IDL, Struts, Content, BPMN, DoDAF, Estructura de components, ...
Llenguatges	Java 5.0, C++, C#, CORBA IDL 3.0, DDL, WSDL 1.1, XML Schema 1.0
Framework	Genera marcs de diversos tipus (DoDAF)
Perfils UML	Si
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java 5.0 (class, jar, zip), C++, EJB 2.0, Java Bytecode, C#, CORBA IDL 3.0, CIL (MSIL), WSDL 1.1, XML Schema 1.0
Diagrames	Classes, Dependència entre paquets, Classes derivades, Dependències, Arbres d'herències Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 5: Característiques de l'eina MagicDraw

Característiques	Metamill UML CASE Tool V4.2
Generals	
URL	http://www.metamill.com
Preu	$125
Patrons	No
Plataformes	Windows, Linux
Extensibilitat	No
Generació de codi	
XMI	Suporta, importa i exporta l'estàndard XMI 1.2
UML	Suporta UML 2.0
Estereotips	Si
Restriccions OCL	Si
Diagrames UML	Classes, Casos d'ús, Paquets, Estats, Activitat, Objectes, Seqüències, Implementació (components i desplegament), Comunicació
Altres tipus de diagrames	Implementació física, Estructura de components
Llenguatges	Java, C++, C#, C
Framework	No
Perfils UML	No
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java, C++, VB.Net, C
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 6: Característiques de l'eina Metamill

Característiques	NetBeans Enterprise 5.5
Generals	
URL	http://www.netbeans.org
Preu	Open-source (llicència LDDL "*Sun Public Licence*") (Eina escrita amb Java)
Patrons	Si
Plataformes	Microsoft, Linux, Solaris
Extensibilitat	Plug-in, Open API
Generació de codi	
XMI	Importa XMI 1.2 per documents MOF 1.4, exporta XMI 1.2
UML	Suporta notacions en UML 2.0
Estereotips	Si
Restriccions OCL	Si
Diagrames UML	Activitat, Classes, Col·laboració, Seqüència, Estats, Casos d'ús, Implementació (components i desplegament)
Altres tipus de diagrames	Seguiment
Llenguatges	Java, C#, C++, CORBA IDL, XML
Framework	Si (JSF, ADF, Dynamic Faces, jMaki, DWR, JSON-RPC)
Perfils UML	Si
Tagged values	No
Reenginyeria inversa	
Llenguatges	Java, C#, C++, CORBA IDL,
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 7: Característiques de l'eina NetBeans

Característiques	Objecteering 6 Enterprise
Generals	
URL	http://www.objecteering.com
Preu	€990
Patrons	GoF, Gamma, ...
Plataformes	Windows, Linux, Solaris
Extensibilitat	Add-ons, Plug-in
Generació de codi	
XMI	Suporta XMI 1.1
UML	Utilitza l'UML 2.0 metamodel
Estereotips	Si
Restriccions OCL	Si
Diagrames UML	UML 2.0 = Col·laboració, Objectes, Classes UML 1.4 = Activitat, Seqüència, Estats, Casos d'ús
Altres tipus de diagrames	Comportament
Llenguatges	Java, C++, SQL, C#, CORBA IDL, Fortran
Framework	Si (JSF, Spring, Struts, ...)
Perfils UML	Si
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java (class, zip, jar), C#, C++
Diagrames	Associacions, Herència, Casos d'ús, Paquets Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 8: Característiques de l'eina Objecteering

Característiques	Poseidon Professional 4.2
Generals	
URL	http://www.gentleware.com
Preu	€699
Patrons	No
Plataformes	Plataforma independent amb JVM
Extensibilitat	Plug-in, API
Generació de codi	
XMI	Suporta, exporta i importa XMI 1.2
UML	Suporta UML 2.0, alguns diagrames solament UML 1.4
Estereotips	Si
Restriccions OCL	Si (pre, post, invariant, inicialització)
Diagrames UML	Estats, Activitat, Seqüència, Classes, Objectes, Casos d'ús, Col·laboració, Implementació (components i desplegament)
Altres tipus de diagrames	Funcionalitats
Llenguatges	C#, C++, CORBA IDL, Delphi, Perl, PHP4, SQL DDL, VB.NET
Framework	Si (JSF, Struts mitjançant plug-ins)
Perfils UML	Si
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 9: Característiques de l'eina Poseidon

Característiques	Rational Rose Enterprise
Generals	
URL	http://www.rational.com
Preu	$4432
Patrons	GoF, POSA, GRASP, J2EE, ELB, ...
Plataformes	Windows, Unix/Linux
Extensibilitat	Add-in, API
Generació de codi	
XMI	Suporta XMI, exporta en format XMI-DTD i XMI[UML]
UML	Suporta UML 1.4 i alguns elements UML 2.0
Estereotips	Si
Restriccions OCL	Si (pre, post, invariant)
Diagrames UML	Classes, Seqüència, Estat, Casos d'ús, Col·laboració, Implementació (components i desplegament)
Altres tipus de diagrames	Magatzem físic / Seguiment, Dades físiques / taules, XML DTD
Llenguatges	Ada, ANSI C++, C++, CORBA IDL, Java, Visual C++, Visual Bàsic
Framework	Si (Swing, JUnit, Servlet, JSP, EJB, Web Services, JDO, JMS, Struts, JSF, ...)
Perfils UML	Si
Tagged values	Si
Reenginyeria inversa	
Llenguatges	Java (class, jar), Ada, ANSI C++, C++, CORBA IDL, Visual C++, Visual Bàsic, COM
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 10: Característiques de l'eina Rational Rose

Característiques	StartUML 5.0
Generals	
URL	http://staruml.sourceforge.net
Preu	Open-source (llicència GPL, "*GNU Public Licence*") (Eina escrita amb Delphi)
Patrons	GoF, EJB, definits per l'usuari
Plataformes	Windows (Win32)
Extensibilitat	Open API, COM-based (plug-in)
Generació de codi	
XMI	Importa i exporta XMI 1.1 per models UML 1.3
UML	Suporta UML 2.0, importa UML 1.3
Estereotips	Si
Restriccions OCL	No
Diagrames UML	Classes, Casos d'ús, Estats, Activitat, Seqüències, Col·laboració, Implementació (components i desplegament)
Altres tipus de diagrames	Definits per l'usuari, ER, Comunicacions, Estructura de components
Llenguatges	Java, C++, C#
Framework	No
Perfils UML	No
Tagged values	No
Reenginyeria inversa	
Llenguatges	Java, C++, C#
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 11: Característiques de l'eina StartUML

Característiques	Sun Java Studio Enterprise 8
Generals	
URL	http://developers.sun.com
Preu	Copia lliure o $10 amb un CD o DVD (Eina escrita amb Java)
Patrons	EJB, GoF
Plataformes	Windows i plataformes amb JVM
Extensibilitat	Open API, codi obert
Generació de codi	
XMI	No importa ni exporta el format XMI
UML	Suporta el model UML 1.4
Estereotips	Si
Restriccions OCL	No
Diagrames UML	Classes, Casos d'ús, Seqüència, Activitat, Estats, Implementació (components i desplegament), Comunicació
Altres tipus de diagrames	No
Llenguatges	Java
Framework	Si (Struts, JSF, IoC, ...)
Perfils UML	No
Tagged values	No
Reenginyeria inversa	
Llenguatges	Java
Diagrames	Control de fluxos, Seqüència (d'operacions) Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 12: Característiques de l'eina Sun Java Studio

Característiques	Visual Paradigm Enterprise 5.3
Generals	
URL	http://www.visual-paradigm.com
Preu	$1399
Patrons	No
Plataformes	Windows, Linux, Mac i totes les plataformes amb JVM
Extensibilitat	API, Plug-in
Generació de codi	
XMI	Importa i exporta fins XMI 1.1 per model UML 1.3 i UML 1.4 respectivament, Suporta XMI 2.1 per models UML 2.0
UML	Suporta UML 2.0, Importar VP-UML
Estereotips	Si
Restriccions OCL	Si
Diagrames UML	Classes, Casos d'ús, Seqüències, Estats, Activitat, Col·laboració, Implementació (components i desplegament), Comunicació
Altres tipus de diagrames	Desplegament, Temps, Interacció, Entitat Relació, CRC Card, BPMN, ORM, Estructura de components
Llenguatges	C#, C++, VB.NET, Java, EJB, DDL
Framework	Si (EOF, DoDAF, ...)
Perfils UML	Si
Tagged values	Si
Reenginyeria inversa	
Llenguatges	XML, XML Schema, .NET dll, Java (class, jar, zip), C++, CORBA IDL, PHP 5.0, ADA9x
Diagrames	Classes Si el codi analitzat té la informació suficient, pot generar els digrames que utilitza per la generació de codi

Taula 13: Característiques de l'eina Visual Paradigm

5.3. Comparació del format XMI

Una de les característiques més rellevants de les eines estudiades és el format XMI que suporten. Mitjançant aquest format es facilita l'intercanvi d'informació i metainformació, es proporciona independència de l'eina sobre el llenguatge i la plataforma utilitzada i simplifica la comunicació entre diferents aplicacions.

XMI

L'especificació XMI defineix els següents aspectes:

- Conjunt de regles de producció de DTD
- Conjunt de regles de producció de documents XMI

El propòsit de XMI és l'intercanvi de metainformació entre diferents eines i aquest integra tres estàndards:

- MOF. Llenguatge per definir llenguatges de modelat
- UML. Llenguatge unificat de modelar
- XML. Estàndard per la definició de llenguatges de marques

L'alineació del metamodel UML 2.0 amb el metamodel MOF 2.0 simplifica l'intercanvi de models via XMI i la interoperabilitat entre diferents eines

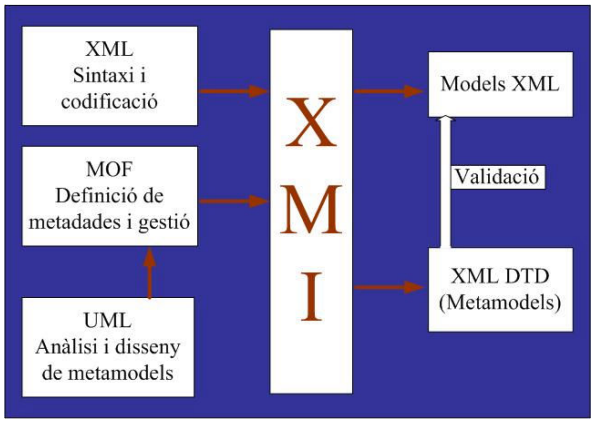


Figura 14: XMI

El nucli d'UML 2.0 i el nucli de MOF 2.0 comparteixen els mateixos elements de metadodel. L'especificació del nucli unificat MOF 2.0 ha d'estar arquitectònicament alineada amb l'infraestructura d'UML

MOF

MOF defineix una arquitectura de llenguatges de modelat que consta de quatre capes:

- Capa M3: MOF.
- Capa M2: UML.
- Capa M1: Model de l'usuari.
- Capa M0: Instàncies en temps d'execució.

UML

UML és el llenguatge gràfic que permet visualitzar, especificar, construir i documentar un sistema. Proporciona el mètode per construir el model del sistema incorporant mecanismes

d'extensió, (estereotips, valors etiquetats i restriccions), que permeten personalitzar-lo per diferents aplicacions i tecnologies.

XML

L'estàndard XML està compost de quatre components

- Documents XML
- XML DTD
- XML parser
- Aplicació XML

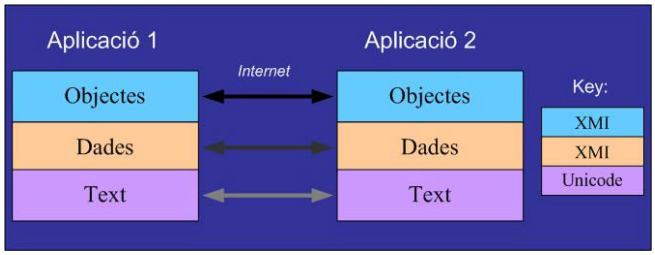


Figura 15: Importació i exportació de XMI

Avantatges i desavantatges

Principals avantatges de XMI:

- Està basat en els principals estàndards industrials. Això significa que està construïda sobre una bona base.
- És independent de la plataforma i de la tecnologia utilitzada. Qualsevol aplicació pot importar i exportar qualsevol fitxer XMI sense que importi la seva procedència.
- Únic format.
- Gran quantitat d'eines suporten aquest format.

Principals desavantatges de XMI:

- No reconeix dades gràfiques
- Incompatibilitat entre versions.
- No totes les eines suporten totalment el format XMI
- És una especificació nova i no està perfectament detallada, per aquest motiu pot haver alguna diferència entre diferents aplicacions

Eines	**Format XMI**
Altova Umodel 2006 r0.22	Importa i suporta fitxers XMI 2.1 Exporta fitxers XMI2.1 per UML 2.0, XMI2.1 per UML 2.1, UUIDs, Umodel Extensions, Pretty-print XMI output
ArgoUML 0.22	Suporta l'estàndard XMI 1.1 amb fitxers que contenen el model UML 1.3 i suporta l'estàndard XMI 1.2 amb fitxers que contenen el model UML 1.4. Exporta XMI 1.2 utilitzant models UML 1.4
JUDE Professional 3.1b2	Suporta, importa i exporta un format original XMI 1.1 amb un model UML 1.4. No pot exportar en XMI el diagrama de components. Alguns diagrames solament poden ser importats o exportats en format XMI 1.1 amb un model UML 1.3. Alguns models s'han de convertir per poder-los exportar en XMI
MagicDraw UML Enterprise 11.6	Suporta XMI 2.1. Fitxers natius estan guardats en XMI format. Es pot guardar amb la màxima informació amb l'opció "Save Rich XMI" S'exporta el model a EMF basat en UML2 (V1.x) compatible amb XMI
Metamill UML CASE Tool V4.2	Suporta models de fitxers amb l'estàndard XMI 1.2. Permet l'accés extern d'eines XML. Importa i exporta amb el XMI 1.2, però el suport està limitat al XMI 1.0

Taula 14A: Característiques XMI de les eines

Eines	Format XMI
NetBeans Enterprise 5.5	Guarda els models MOF en fitxers XMI. Carrega els fitxers metamodels XMI dins el MDR i genera una API per després accedir-hi. Canvia les metadades i els metamodels mitjançant les especificacions XMI
Objecteering 6 Enterprise	Suporta , importa i exporta XMI 1.1
Poseidon Professional 4.2	Suporta, exporta i importa XMI 1.2, Utilitza el XMI com a format per defecte
Rational Rose Enterprise	Suporta XMI, exporta en format XMI-DTD i XMI[UML]
StartUML 5.0	Suporta XMI 1.1. Importa i exporta XMI 1.1 per models UML 1.3 (generant UUID, Timestamp, diagrames i vistes d'elements) importa i exporta XMI 1.1 per models UML 1.3 amb extensions Rose (generant UUID, Timestamp, diagrames i vistes d'elements)
Sun Java Studio Enterprise 8	No importa ni exporta el format XMI
Visual Paradigm Enterprise 5.3	Importa i exporta XMI 1.1 per models UML 1.3 (Unisys extension) i models UML 1.4 (OMG), també pot exportar XMI 2.1 per models UML 2.0

Taula 14B: Característiques XMI de les eines

6. Realització d'una aplicació

6.1. Descripció

Aquest programari implementa la gestió del catàleg de productes de l'empresa PFC. Permet al personal de PFC gestionar el seu catàleg de productes, tenint en compte que els productes del catàleg s'organitzen en categories.

El programari pot fer altes, baixes, consultes, cerques i modificacions de productes del catàleg. També pot llistar totes les categories del catàleg de productes i tots els productes que hi ha dins d'una categoria determinada.

6.2. Anàlisi

Es una petita part d'un programari J2EE de comerç electrònic. Concretament s'implementa una part de la gestió d'un catàleg de productes on-line.

És un programari que s'implementa de manera distribuïda pel fet de ser una aplicació que ha de ser molt escalable.

6.2.1. Casos d'ús

Es pot distingir un sol actor que és l'administrador i els següents casos d'ús:

- Entrar al sistema
- Llistar categories
- Llistar productes
- Cercar productes
- Alta producte
- Modificar producte
- Consultar producte
- Baixa producte
- Alta categoria
- Baixa categoria

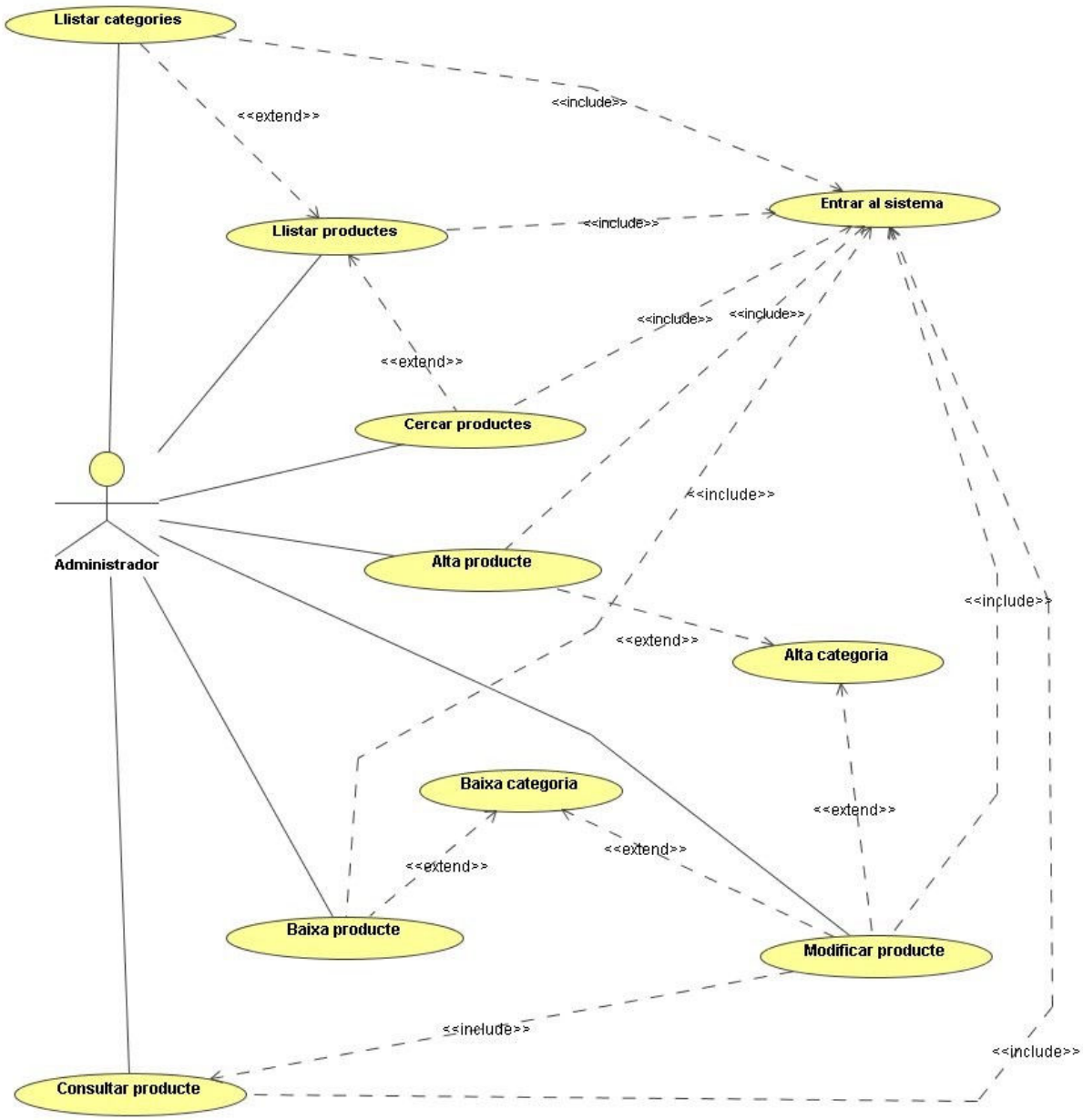


Diagrama 2: Diagrama de casos d'ús

6.2.2. Descripció dels casos d'ús

A continuació es resumeixen les principals característiques dels casos d'ús del programari

Cas d'ús "Entrar al sistema"

Resum funcionalitat general: Permet als usuaris amb rol d'administrador identificar-se al sistema.

Paper dins el treball de l'usuari: Primera tasca a realitzar al entrar al sistema.

Actors: Administrador

Casos d'ús relacionats: Cap

Precondició: L'usuari ha de tenir un nom d'usuari i una paraula clau que el sistema reconegui.

Postcondicio: L'usuari entra al sistema com administrador reconegut

Descripció: Permet identificar un usuari com administrador del sistema. Ha d'introduir el nom d'usuari i la paraula clau que té assignada prèviament.

Alternatives de procés i excepcions: Si l'usuari no és reconegut per el sistema, li presenta una pantalla d'error i no el deixa entrar.

Observacions: El nom d'usuari i la paraula clau són subministrades a l'administrador amb anterioritat a l'entrada al sistema.

Cas d'ús "Llistar categories"

Resum funcionalitat general: Permet als administradors llistar les categories que té el catàleg.

Paper dins el treball de l'usuari: És un cas d'ús per fer llistats.

Actors: Administrador

Casos d'ús relacionats: Entrar al sistema

Precondició: Haver estat identificat pel sistema. Que hi hagin categories en la base de dades.

Postcondició: El sistema presenta a l'administrador el llistat demanat

Descripció: El sistema presenta a l'administrador totes les categories en que estan agrupats els productes.

Alternatives de procés i excepcions: Cap

Observacions: Les categories són gestionades internament. En cas de que no hi hagi cap categoria es presentarà la llista buida.

Cas d'ús "Llistar productes"

Resum funcionalitat general: Permet als administradors llistar els productes que hi ha al catàleg.

Paper dins el treball de l'usuari: És un cas d'ús per fer llistats.

Actors: Administrador

Casos d'ús relacionats: Cercar productes, Entrar al sistema

Precondició: Haver estat identificat pel sistema. Que hi hagin productes en la base de dades.

Postcondició: El sistema ha presentat a l'administrador el llistat demanat

Descripció: El sistema presenta a l'administrador tots els productes que hi ha al catàleg.

Alternatives de procés i excepcions: Cap

Observacions: En cas de que no hi hagi cap producte es presentarà la llista buida.

Cas d'ús "Cercar productes"

Resum funcionalitat general: Permet als administradors buscar productes.
Permet fer la cerca dels productes del catàleg especificant el nom d'un producte.

Paper dins el treball de l'usuari: És un cas d'ús de cerca.

Actors: Administrador

Casos d'ús relacionats: Entrar al sistema, Llistar productes

Precondició: Haver estat identificat pel sistema. L'administrador ha de conèixer el nom del producte a buscar.

Postcondició: El sistema presenta a l'administrador el llistat de tots els productes en que el seu nom coincideixi amb el nom cercat.

Descripció: Permet als administradors buscar tots els productes catalogats amb un nom determinat i llistar-los

Alternatives de procés i excepcions: Cap

Observacions: En cas de que no hi hagi cap producte amb el nom cercat es presentarà la llista buida.

Cas d'ús "Alta producte"

Resum funcionalitat general: Permet als administradors introduir nous productes al catàleg

Paper dins el treball de l'usuari: És el cas d'ús introducció d'un producte a la base de dades.

Actors: Administrador

Casos d'ús relacionats: Alta categoria

Precondició: Haver estat identificat pel sistema. El nou producte no ha d'existir en el catàleg

Postcondició: El sistema guarda les dades del nou producte. Si en el catàleg no existia la categoria del nou producte es donarà d'alta la nova categoria

Descripció: Permet afegir un nou producte al catàleg. En aquesta acció l'administrador haurà d'introduir el codi, nom, descripció, categoria i preu del nou producte

Alternatives de procés i excepcions: En cas de que el producte ja existeixi a la base de dades es presentarà una pantalla d'error

Observacions: Com a mínim s'ha de introduir el codi del nou producte i la categoria a que pertany

Cas d'ús "Consulta producte"

Resum funcionalitat general: Permet als administradors consultar les característiques d'un producte determinat

Paper dins el treball de l'usuari: És el cas d'ús més utilitzat del programari.

Actors: Administrador

Casos d'ús relacionats: Llistar productes, Modificar producte

Precondició: Haver estat identificat pel sistema. L'administrador ha de conèixer el producte prèviament per poder-lo consultar.

Postcondició: El sistema presenta a l'administrador el detall d'un producte.

Descripció: Permet visualitzar totes les dades referents a un producte. En aquesta acció l'administrador podrà consultar el codi, nom, descripció, categoria i preu del producte

Alternatives de procés i excepcions: En cas de que el producte no existeixi a la base de dades es presentarà una pantalla d'error

Cas d'ús "Modificar producte"

Resum funcionalitat general: Permet als administradors modificar les característiques dels productes

Paper dins el treball de l'usuari: És un dels casos d'ús més utilitzats.

Actors: Administrador

Casos d'ús relacionats: Alta categoria, Baixa categoria

Precondició: Haver estat identificat pel sistema. El producte ha d'existir a la base de dades, l'administrador ha de conèixer el producte prèviament.

Postcondició: El sistema modifica els atributs del producte de la base de dades. Si es modifica la categoria del producte i en el catàleg no existeix la categoria del producte modificat es donarà d'alta una nova categoria i, si en el catàleg no existeix cap més producte de la categoria modificada es donarà de baixa la categoria anterior.

Descripció: Permet modificar les dades d'un producte. En aquesta acció l'administrador podrà veure les dades actuals del producte i podrà modificar el codi, nom, descripció, categoria i preu del producte.

Alternatives de procés i excepcions: En cas de que el producte no existeixi a la base de dades es presentarà una pantalla d'error

Cas d'ús "Baixa producte"

Resum funcionalitat general: Permet als administradors descatalogar un producte

Paper dins el treball de l'usuari: És un cas d'ús poc utilitzat del programari.

Actors: Administrador

Casos d'ús relacionats: Baixa categoria

Precondició: Haver estat identificat pel sistema. El producte ha d'existir a la base de dades i l'administrador ha de conèixer el producte prèviament.

Postcondició: Esborra de la base de dades el registre del producte. Si en el catàleg no existeix cap més producte de la mateixa categoria modificada es donarà de baixa la categoria anterior.

Descripció: Permet esborrar del catàleg un producte. En aquesta acció l'administrador haurà d'introduir el codi del producte a donar de baixa.

Alternatives de procés i excepcions: En cas de que el producte no existeixi a la base de dades es presentarà una pantalla d'error

Cas d'ús "Alta categoria"

Resum funcionalitat general: Introdueix noves categories al catàleg

Paper dins el treball de l'usuari: És el cas d'ús d'execució automàtica.

Actors: Administrador

Casos d'ús relacionats: Altra producte, Modificar producte

Precondició: La categoria no ha d'existir a la base de dades

Postcondició: El sistema guarda la nova categoria a la base de dades

Descripció: El sistema al detectar que la categoria no existeix al catàleg l'introdueix com a nova categoria. Aquesta acció l'executarà el sistema

Alternatives de procés i excepcions: Cap

Cas d'ús "Baixa categoria"

Resum funcionalitat general: Esborra categories del catàleg

Paper dins el treball de l'usuari: És el cas d'ús d'execució automàtica.

Actors: Administrador

Casos d'ús relacionats: Baixa producte, Modificar producte

Precondició: La categoria ha d'existir a la base de dades

Postcondició: El sistema el registre de la categoria de la base de dades

Descripció: El sistema al detectar una categoria sense cap producte, la descataloga. Aquesta acció l'executarà el sistema

Alternatives de procés i excepcions: Cap

6.2.3. Diagrama de classes del model conceptual

Es pot veure que hi ha quatre classes. Tres d'elles (Catàleg, Categoria, Producte) són les que gestionarà el programari i la classe Administrador la gestionarà l'arquitectura J2EE.

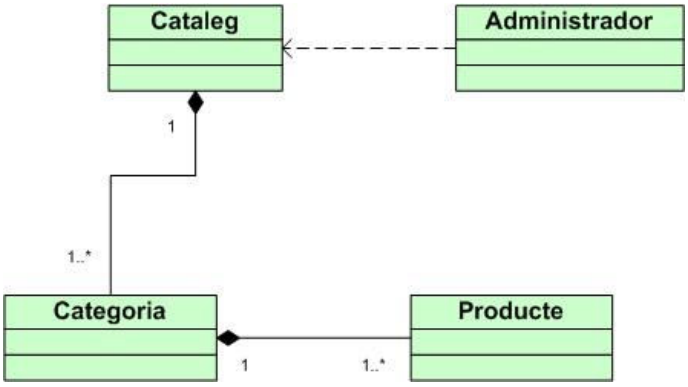


Diagrama 3: Diagrama de classes

6.3. Disseny

Al programari s'accedirà mitjançant un navegador

S'utilitzarà el patró MVC per separar les dades del programari. El navegador enviarà peticions, i un servlet s'encarregarà d'invocar la lògica de negoci pertinent.

L'estructura del programari utilitzarà JSP, Servlets, EJB de sessió, EJB d'entitat i Taules de base de dades. S'utilitzaran els EJB sense estat, els EJB d'entitat seran CMP, i la connexió entre els EJB d'entitat i la base de dades es farà mitjançant el connector JDBC.

Construirem l'aplicació fent servir una arquitectura J2EE multicapa i distribuïda de la següent manera:

Capes lògiques:
- Client
- Presentació
- Negoci
- Dades

Capes físiques:
- Client (client web remot)
- Presentació i negoci (servidor d'aplicacions)
- Dades (servidor de dades)

Les màquines físiques en que s'instal·larà el programari són:
- Maquina client. Té instal·lat un navegador HTTP
- Servidor d'aplicacions. Té el servidor d'aplicacions JBoss instal·lat en un contenidor web i en un contenidor d'EJBs. Aquest servidor es comunica amb la base de dades relacional mitjançant JDBC.
- Màquina de dades. Té el servidor de base de dades relacional MySQL que emmagatzema les dades persistents de l'aplicació.

Els administradors del catàleg hi podran accedir mitjançant el nom d'usuari i una paraula clau

Gestió del productes

L'administrador a l'hora de gestionar qualsevol producte haurà d'introduir com a mínim el codi del producte i la categoria del producte

6.3.1. Diagrama de components

Els components més importants es poden veure al diagrama de components d'alt nivell, on tindrem el component *Catàleg* a la capa de negoci que oferirà la seva interfície *ICataleg* als clients de la capa de presentació. També mostrem el component de *Seguretat* que permet que solament puguin entrar al sistema els usuaris que s'hagin validat prèviament.

El component de seguretat ens el proporcionarà la plataforma J2EE i no caldrà que l'implementem

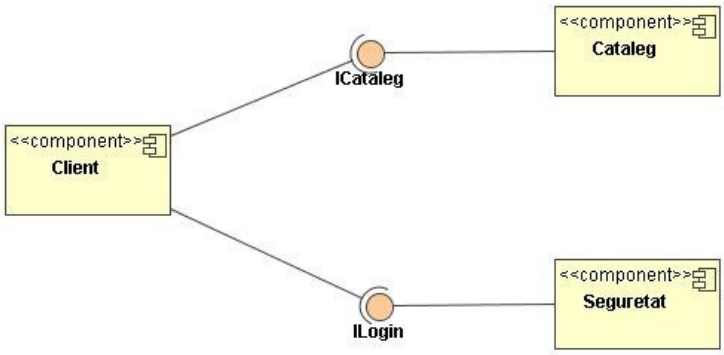


Diagrama 4: Diagrama de components d'alt nivell

Els components els podem veure al diagrama de components següent que reflecteix amb més detall els components del programari.

Hi ha el component *CatalegBean* a la capa de negoci i ofereix les seves interfícies *Catàleg* i *CatalegHome*. També mostra els components *ProducteBean* i CategoriaBean a la capa de persistència i ofereix les seves interfícies *ProducteHome, Producte, ProducteLocal, ProducteLocalHome, CategoriaHome*, *Categoria, CategoriaLocal, CategoriaLocalHome.* Podem veure els components *ControladorServlet* i *JSP* a la capa de presentació i per últim el component *HTML* a la capa client.

Es poden veure on estaran situats els components dins de cada mòdul.

En el següent diagrama la base de dades s'ha representat en un mòdul separat però en el nostre programari estarà situat en el servidor de dades

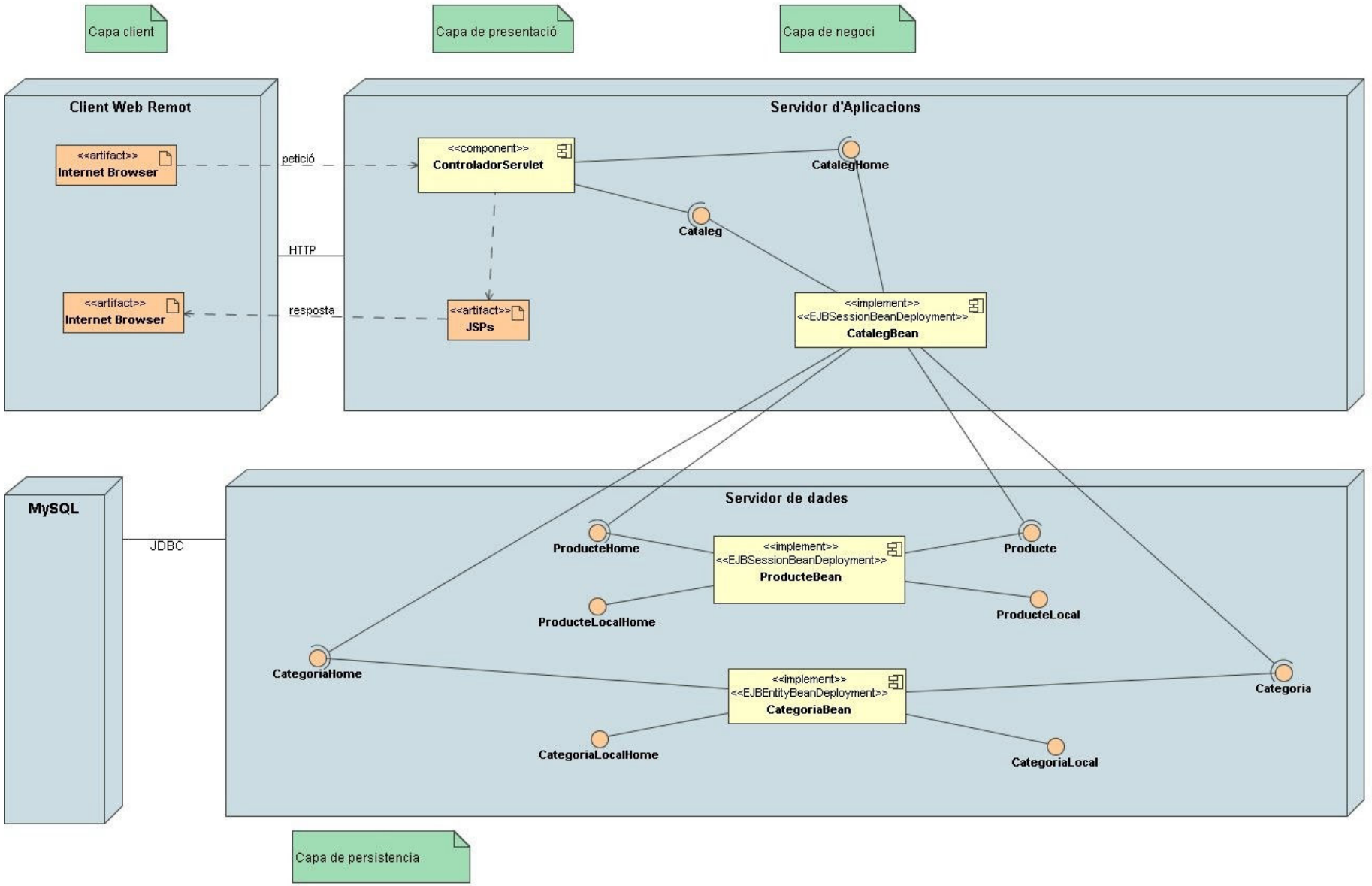


Diagrama 5: Diagrama de components detallat

En la nostra aplicació els mòduls *servidor d'aplicacions* i *servidor de dades* estaran gestionats per JBoss, mentre que el mòdul MySQL estarà gestionat per MySQL.

6.3.2. Diagrama de classes de disseny

S'han representat separadament les classes client del diagrama general, ja que són classes que utilitzarem per provar el funcionament del sistema i no per el funcionament del programari complert.

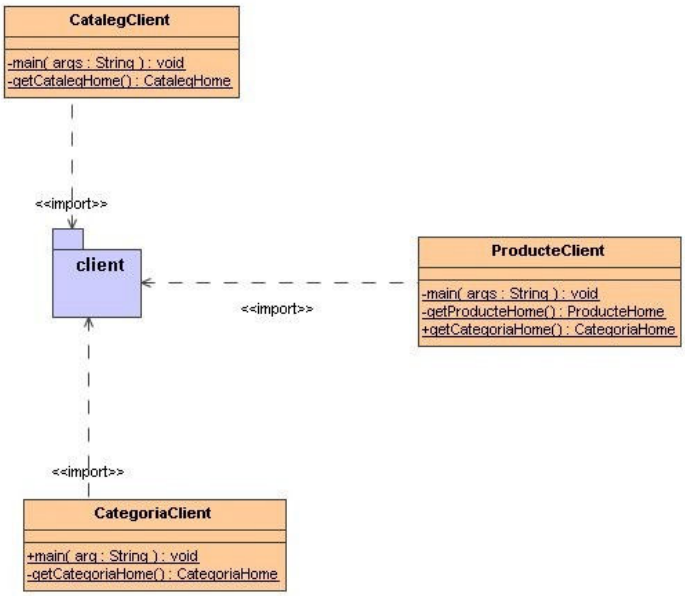


Diagrama 6: Diagrama de classes de prova

A continuació presentem el principal diagrama de classes del projecte, on es representen les classes que intervenen en el programari

Es poden agrupar en quatres grans grups:

- Interfícies
  - Inici (Ex: ProducteHome)
  - Local Inici (Ex: ProducteLocalHome)
  - Remot (Ex: Producte)
  - Local Remot (Ex: ProducteLocal)

- Contenidors EJB (Ex: ProducteBean)
- Value Objects (Ex: ProducteVO)
- Servlet (Ex: ControladorServlet)

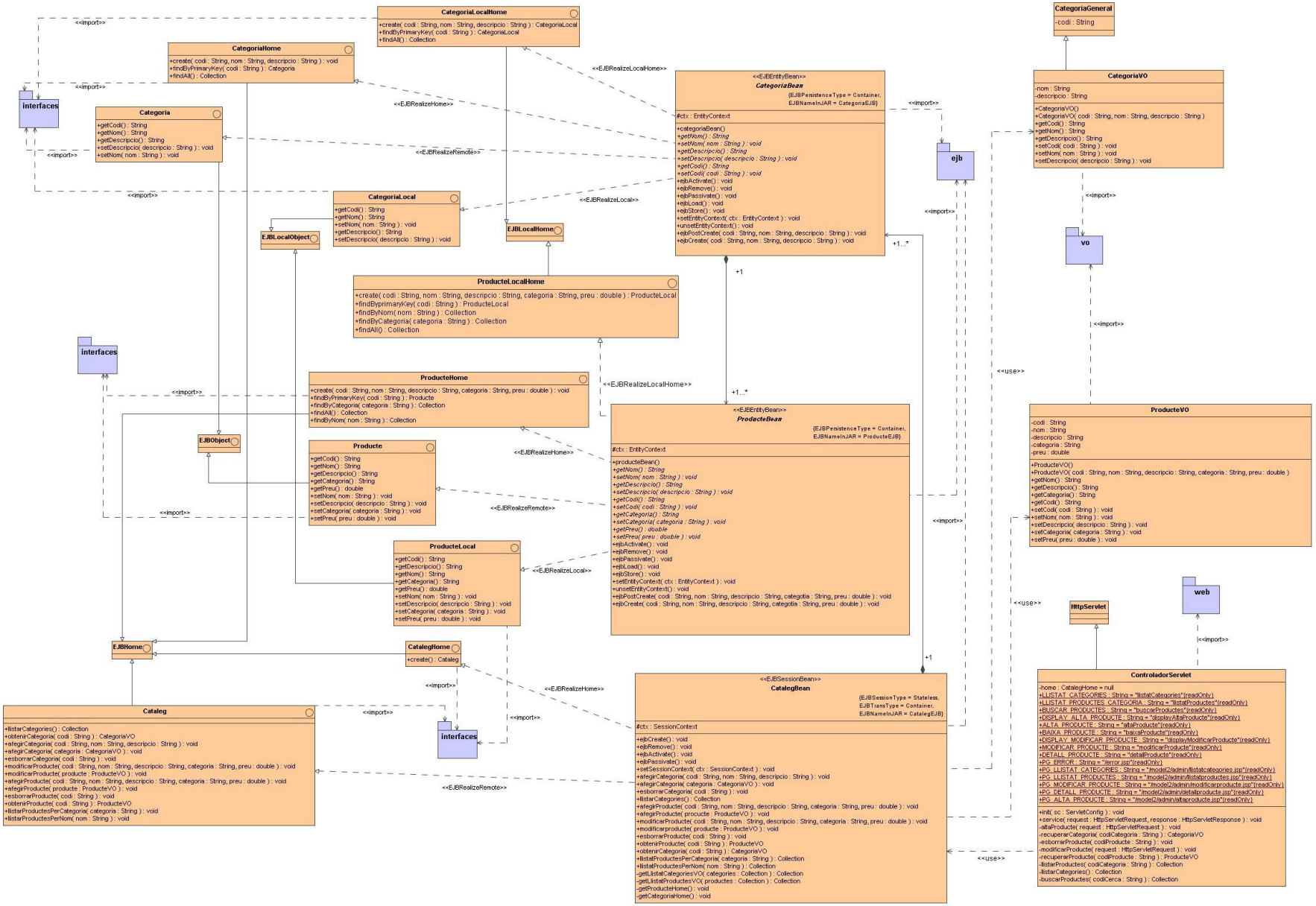


Diagrama 7: Diagrama de classes principal (MagicDraw)

6.3.3. Classes

- CatalegBean
- CategoriaBean
- ProducteBean
- Catàleg
- Categoria
- Producte
- CatalegHome
- CategoriaHome
- ProducteHome
- CategoriaLocal
- ProducteLocal
- CategoriaLocalHome
- ProducteLocalHome
- ControladorServlet

6.3.4. Diagrama de l'estructura de la base de dades

En el següent diagrama podem comprovar els valors mínims que l'administrador ha d'introduir a l'hora de donar d'alta un nou producte al catàleg. Aquest atributs són: el codi del producte i la categoria a que pertany

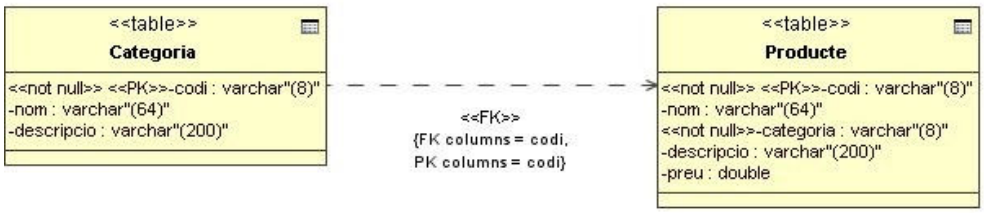


Diagrama 8: Diagrama de taules de BD

Con ja hem comentat l'objecte Administrador el gestionarà l'arquitectura J2EE. Per aquest motiu no esta inclòs en aquest diagrama

6.3.5. Interfícies gràfiques

Les interfícies gràfiques d'usuari seran presentades per el navegador i enviades per el Servlet en forma de pàgines JSP. Tindran el format següent:

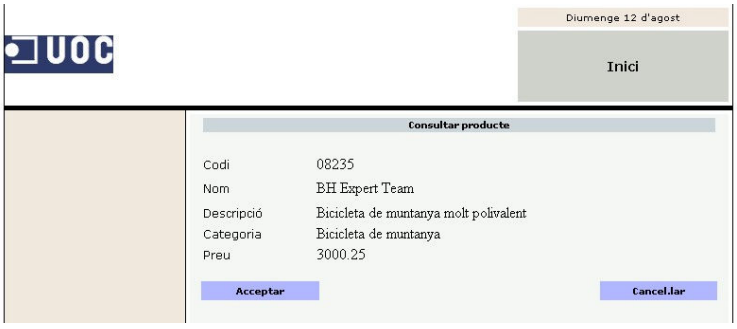


Figura 16: Pantalla "Consulta producte"

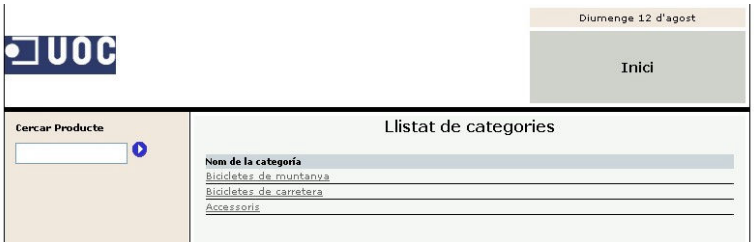


Figura 17: Pantalla "Llistar Categories"

Figura 18: Pantalla "Llistar productes"

6.3.6. Observacions en el disseny

Hi han algunes consideracions més a tenir en compte en el disseny d'aquest programari.

S'utilitzarà el EJB *CategoriaBean* que realitzarà el mapeig a la taula Categoria i el EJB *ProducteBean* que realitzarà el mapeig a la taula Producte. Aquestes dues EJBs faran servir CMP. D'aquesta manera la persistència la realitzarà el contenidor.

S'implementaran interfícies remotes i locals

El EJB *CatalegBean* s'implementarà sense estat ja que no cal mantenir cap estat entre diferents crides dels clients

Els Value Objects seran objectes serialitzables per poder-los transmetre per la xarxa

6.4. Disseny amb les eines

Les eines seleccionades per analitzar el codi que generen són: MagicDraw i ArgoUML. S'han escollit aquestes dues eines per les següents raons:

MagicDraw. Per ser la millor després de realitzar l'estudi de les eines escollides.

ArgoUML. Per que és de lliure distribució, està realitzada en java, és molt diferent de MagicDraw i el que volem veure són diferències en el resultat.

6.4.1. Disseny amb MagicDraw

Per poder generar el codi automàticament amb l'eina MagicDraw del programari de "gestió de productes d'un catàleg" s'han realitzat els diagrames presentats en els apartats anteriors.

Per poder comprovar alguns aspectes que no estaven inclosos en el programari s'han modificat algunes especificacions per poder fer les comprovacions del codi que genera l'eina.

A més s'han realitzat els següents diagrames, un per la correcta implementació de l'aplicació i l'altra per provar la generació de codi d'un diagrama de seqüència.

6.4.1.1. Diagrama de implementació EJB

En el següent diagrama podem veure el fitxers d'implementació que necessita el programari pel seu funcionament i que tots els components quedin enllaçats

Con que l'administrador és gestionat per l'arquitectura J2EE en el següent diagrama apareixen els components *roles.properties* i *users.properties* que contenen les dades per poder gestionar els administradors. El component *web* és un descriptor i *ejb-jar* és un descriptor de desplegament per l'arquitectura J2EE. Els components *ProducteBean, CategoriaBean* i *CatalegBean* són contenidors EJB

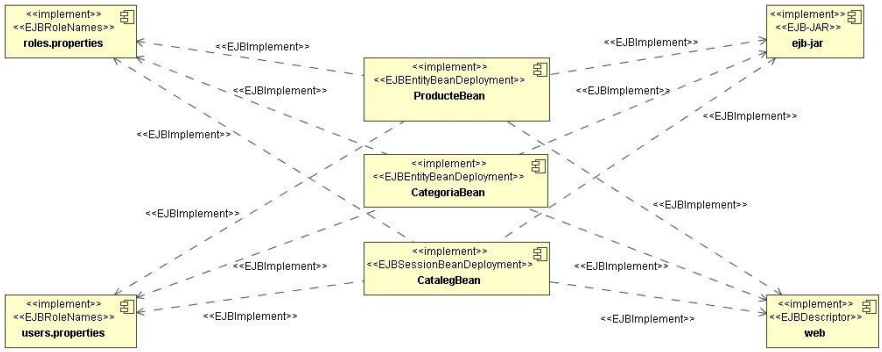


Diagrama 9: Diagrama de implementació

6.4.1.2. Diagrama de seqüència de "Alta producte"

S'ha realitzat el diagrama de seqüència que es pot veure a continuació esperant que MagicDraw generi un codi semblant al següent:

```
.....
altaProducte(request);
.....

private void altaProducte(HttpServletRequest request) throws  RemoteException,
                        CreateException {

        Cataleg cat = home.create();
        ProducteVO prod = new ProducteVO();
        prod.setCodi(request.getParameter("codi"));
        prod.setNom(request.getParameter("nom"));
        prod.setCategoria(request.getParameter("categoria"));
        prod.setDescripcio(request.getParameter("descripcio"));
        prod.setPreu(request.getParameter("preu"));

        cat.afegirProducte(prod);
}
……
```

Figura 19: Codi del mètode "altaProducte" de la classe ControladorServlet

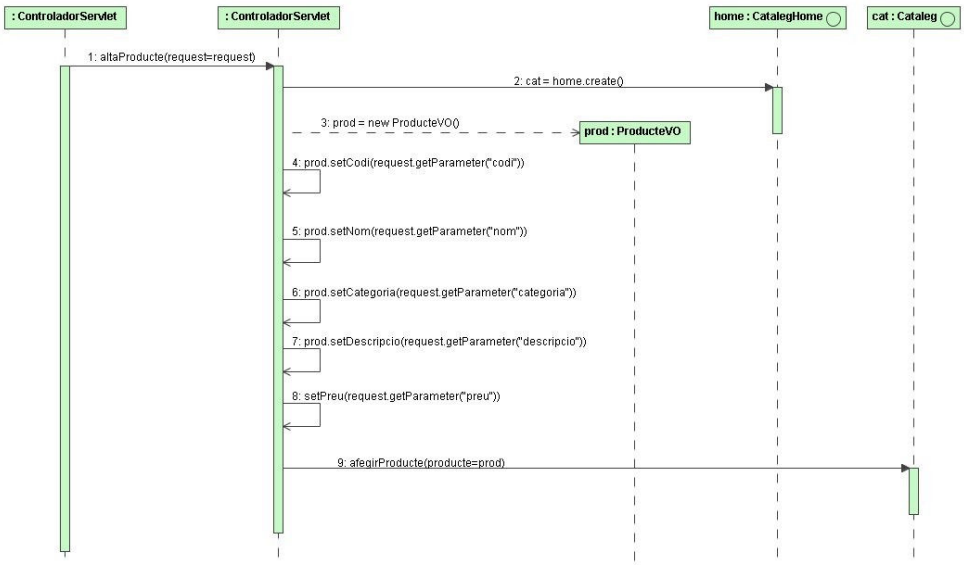


Diagrama 10: Diagrama de seqüència de "Alta producte"

6.4.2. Disseny amb ArgoUML

Per poder generar el codi automàticament amb l'eina ArgoUML del programari de "gestió de productes d'un catàleg", s'ha realitzat el diagrama de casos d'ús i el diagrama de classes amb les mateixes especificacions que s'han utilitzat en el cas de l'eina MagicDraw.

El diagrama de casos d'ús realitzat amb ArgoUML no s'ha plasmat en aquesta memòria ja que és igual al realitzat amb MagicDraw.

6.4.3. Diagrama de classes

En el següent diagrama podem veure el diagrama de classes del programari "gestió de productes d'un catàleg", amb les mateixes modificacions que s'han realitzat amb MagicDraw. D'aquesta manera es podrà realitzar la comparació ja que les especificacions seran les mateixes per les dues eines.

Entre el diagrama de classes que s'ha realitzat amb ArgoUML hi trobem algunes diferencies respecte el diagrama de classes de MagicDraw. Degut a mancances amb l'eina que és més senzilla, li manquen algunes característiques que MagicDraw té implementades.

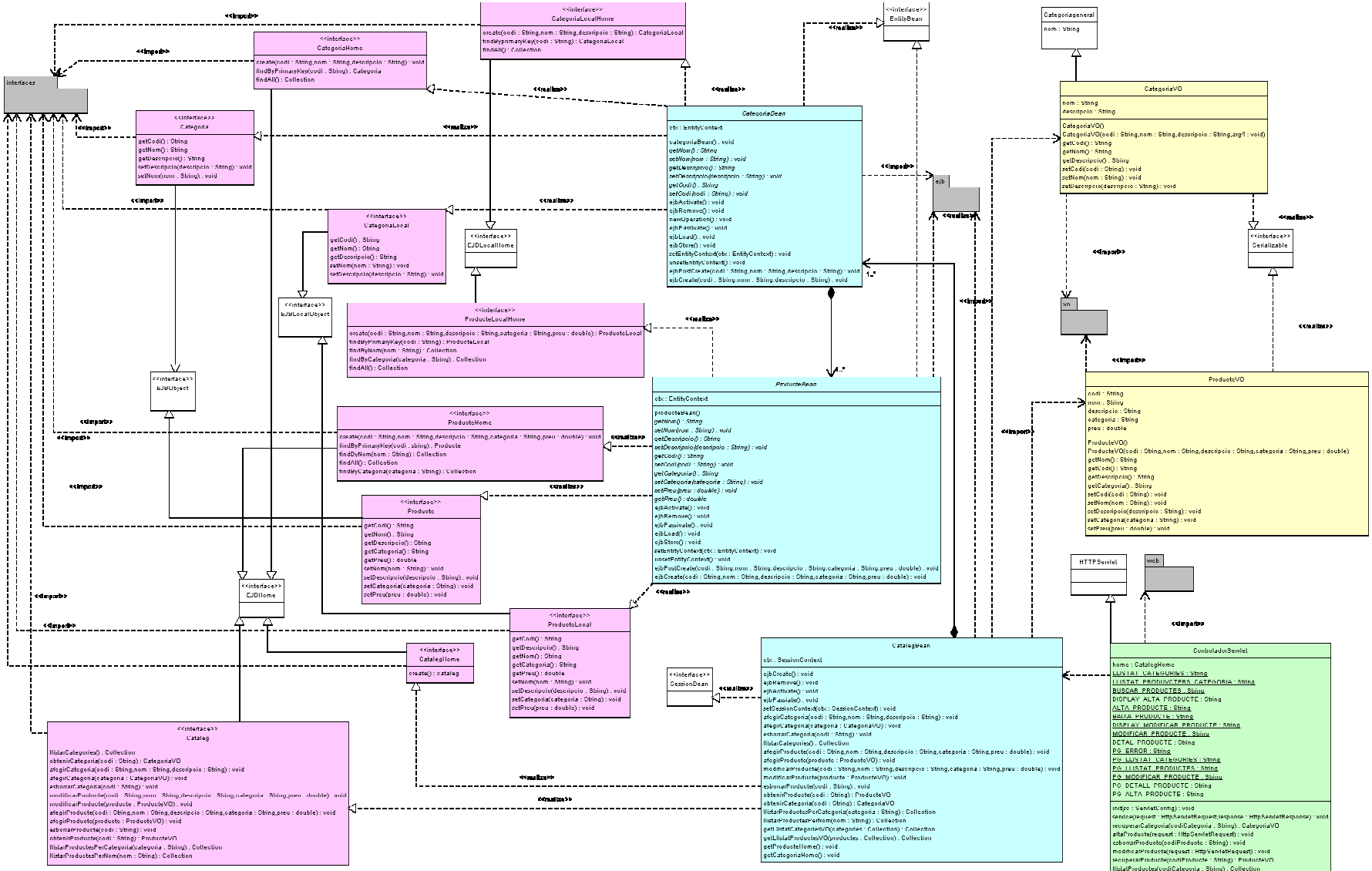


Diagrama 11: Diagrama de classes principal (ArgoUML)

6.5. Generació automàtica de codi

A tall d'exemple en l'annex A hi ha el codi generat automàticament per l'eina MagicDraw de la classe CategoriaBean. En l'annex B hi ha el codi generat automàticament per l'eina ArgoUML de CategoriaBean. D'aquesta manera es poden veure les diferencies.

Atributs:

Dels atributs s'especifica la seva visibilitat (públics, privats, protegits o paquet) i els modificadors d'atributs (estàtica, lectura, leaf).

Comentaris:

Genera els comentaris necessaris per ajudar als programadors a seguir el codi generat. Aquests comentaris no són generats automàticament si no que han de ser introduïts amb anterioritat

Classes:

Té en compte la seva visibilitat, l'herència, les implementacions d'interfaces, classes abstractes i estereotips

Al començament, importa classes de paquets existents i fins i tot importa paquets de classes senceres.

També genera paquets de classes d'interface mitjançant la paraula clau packages.

Mètodes:

Dels mètodes s'especifica la seva visibilitat (públics, privats, protegits o paquet) i dels modificadors dels mètodes (estàtica, arrel, lectura, query, leaf, abstracta). Posa els tipus de paràmetres i el tipus de retorn. Si el tipus de retorn no és void, s'introdueix en el mètode una línea amb el retorn a null.

Relacions:

Les dues eines tracten de manera diferenciada les relacions.

La relació d'agregació del tipus composició, en cas de MagicDraw, genera dos fitxers de la interfície local de les classes que en el nostre cas són: *CategoriaLocal* i *ProducteLocal* en un subdirectori anomenat datatypes. En cas d'ArgoUML, no el té en compte.

La cardinalitat de la relació d'agregació del tipus composició, en el cas de MagicDraw, genera un atribut i en el cas d'ArgoUML genera un vector.

La relació d'ús, les eines de MagicDraw i ArgoUML generen la crida del mètode de que en fa ús.

En la relació d'herència, totes dues eines generen correctament la crida de la classe pare.

En el cas de la implementació d'interfaces, l'eina MagicDraw, ho genera correctament en canvi l'eina ArgoUML ho fa incorrectament degut a que no té en compte l'estructura EJB i ho genera com una classe normal.

Directoris:

Genera l'estructura de directoris que nosaltres hem especificat amb anterioritat i que l'aplicació necessita per funcionar.

Scripts:

Genera automàticament els fitxers necessaris per la creació de taules de bases de dades.

Fitxers de desplegament:

Crea els fitxers de desplegament per el funcionament de l'aplicació.

Excepcions:

En el cas d'alguna excepció d'una classe o mètode, l'eina MagicDraw, genera correctament la crida del mètode de tractament de l'excepció. Amb l'eina ArgoUML no genera la crida, solament ho deixa com a comentari.

Diagrama de seqüència:

Les dues eines no tenen en compte els diagrames de seqüència realitzats per generar codi intern a mètodes.

En la següent taula s'enumeren alguns dels punts a tenir en compte en el codi generat per les eines MagicDraw i ArgoUML.

Punts d'atenció	Codi que l'eina genera correctament	
	ArgoUML 0.22	MagicDraw UML Enterprise 11.0
Diagrama de seqüència	No	No
Cardinalitat de les relacions	Incorrectament	Incorrectament
Relació d'agregació (composició)	Incorrectament	Incorrectament
Relació d'herència	Si	Si
Classe abstracta	Si	Si
Relació de dependència (ús)	Si	Si
Implementació d'interface	Si (1)	Si
Importar classes d'un packet	Si	Si
Adició de classes d'un paquet	Si	Si
Visibilitat d'atributs	privat / protegit / públic / paquet	privat / protegit / públic / paquet
Modificadors d'atributs	estàtica	estàtica / lectura / leaf
Visibilitat de mètodes	privat / protegit / públic / paquet	privat / protegit / públic / paquet
Modificadors de mètodes	estàtica / arrel / query / leaf / abstracta	estàtica / lectura / query / leaf
Excepcions que llança els mètodes	Incorrectament	Si
Documentació	Si	Si

Taula 15: Punts a tenir en compte del codi generat

(1) Implementa d'una manera bàsica les interfícies però no té en compte que és un programari EJB

7. Conclusions

7.1. Conclusions de l'estudi d'eines

Finalment després d'estudiar i provar les eines de generació automàtica de codi més representatives és important assenyalar els beneficis a nivell qualitatiu i productiu que s'obtenen de la utilització d'aquestes eines

S'ha de destacar la diversitat en el format XMI, UML, llenguatges suportats i el preu de compra.

La majoria tenen versions per totes les plataformes i suporten la majoria de diagrames UML.

Totes les eines suporten la reenginyeria inversa i els diagrames que poden generar són principalment els mateixos que en la generació de codi.

Les eines que destaquen més sobre les altres són: MagicDraw UML Enterprise 11.6, VisualParadigm Enterprise 5.3, Rational Rose Enterprise i, com a Open-source, NetBeans Enterprise 5.5

Cal destacar que la millor eina estudiada i més ben documentada ha estat MagicDraw UML Enterprise 11.6

El format XMI és el punt de trobada entre totes les aplicacions sigui quina sigui la tecnologia, plataforma i llenguatge emprat.

El format XMI és molt innovador. Les versions més antigues tenen problemes de compatibilitat entre els diferents estàndards que l'integren, fins i tot, hi ha incompatibilitat entre diferents versions XMI. En les versions més actuals, com les XMI 2.x estan resolts aquests problemes.

Les eines estudiades suporten diferents versions d'importació i l'exportació i com a conseqüència, no són totalment compatibles entre elles. Per que siguin totalment compatibles necessiten actualitzar-se, suportant totalment els formats XMI 2.x

7.2. Conclusions de l'aplicació de MagicDraw i ArgoUML

Finalment després de fer els diagrames necessaris per poder analitzar el codi generat automàticament per les eines MagicDraw i ArgoUML, i partint com a base de la realització d'una part de l'aplicació de la gestió de productes d'un catàleg, es pot determinar que el codi generat prové principalment del diagrama de classes, i que els altres diagrames determinen petits aspectes en el codi generat.

Respecte al codi generat es dedueix que les eines generen un esquelet del programari i que després el programador haurà de completar.

L'esquelet generat és molt complert, però falta tota la programació interna del mètodes que a la fi és un 80% del pes de la programació.

Cal destacar la diferencia que hi ha entre les dues eines en que s'ha realitzat l'aplicació de gestió de productes d'un catàleg. En el cas de l'eina MagicDraw ja té implementats tots els estereotips necessaris per aplicacions actuals i en el cas de l'eina ArgoUML el treball és molt més costos

Un dels aspectes més importants a tenir en compte és que l'ús d'eines MDA de generació de codi pot representar la reutilització de models com a estàndard per l'empresa programadora

També la utilització d'aquestes eines facilita la feina que tenen els programadors per entendre l'esquema general del programari en el que han de treballar.

Com a conclusió, enumerar alguns avantatges en la utilització d'eines MDA. Permeten una major abstracció del programari a realitzar, s'augmenta la productivitat gràcies a l'aplicació de transformació automàtica entre models, millora de la portabilitat ja que els models no estan lligats a cap maquinari, facilita el manteniment del programari construït i facilita la documentació ja que els models conceptuals descriuen realment el sistema.

8. Línies futures de treball

El camp de la generació automàtica de codi està en les primeres fases i per poder anar endavant s'ha de tenir una bona base en la que tots treballen conjuntament, per aquest motiu és de vital importància que totes les empreses utilitzin els estàndards existents.

Les principals línies futures de treball s'haurien de centrar en:

- ✓ Generació del codi intern dels mètodes.
- ✓ Estandarditzar els formats utilitzats per les eines

9. Glossari

- API. Application Programming Interface. Interface de programació d'aplicacions
- AVK. Java Aplication Verification Kit
- BPEL. Business Process Execution Language. Llenguatge d'execució de processos de negoci
- BPMC. Business Process Modeling Notation
- CMP. Persistència gestionada pel contenidor
- Constraints. Restriccions
- EJB. Enterprise Java Beans. API que forma part del estàndard de construcció d'aplicacions empresarials J2EE.
- EOF. Enterprise Object Framework
- EMF. Enhanced Metafile Format
- Frameworks. Marcs
- HTML. HyperText Markup Language. Llenguatge de marques hipertextuals.
- IDE. Entorn de desenvolupament integrat
- J2EE. Java 2 Platform, Enterprise Edition
- JDBC. Java Database Connectivity. API que permet l'execució d'operacions sobre bases de dades des de Java.
- JSP. Java Pages Server. Tecnologia Java que permet generar continguts dinàmics en documents HTML.
- JVM. Java Virtual Machine
- MDA. Model Driver Architecture
- MDR. Meta Data Repository
- MDD. Model Driven Development
- Mind Map. Diagrama utilitzat per representar paraules, idees i tasques,
- MOF. Meta Object Facility
- MVC. Model, vista, controlador. Patró d'arquitectura del software que separa les dades de l'aplicació
- OCL. Object Constraint Language. Llenguatge per la descripció formal d'expressions en el models UML
- OMG. Object Management Group
- Package. Paquet java
- Patterns. Patrons
- Profile. Perfils
- Script. Fitxers d'execució automàtic seqüencial
- Servlet. Objecte que s'executa en un servidor o contenidor J2EE.
- SOA. Service-oriented Architecture.
- UML. Unified Modeling Language. Llenguatge de modelat de sistemes
- Value Objects. Objectes java plans
- XMI. XML d'intercanvi de metadades
- XML. Extensible Markup Language. Llenguatge de marcar extensible

10. Bibliografia

- **Mastering XMI Java Programming With XMI, XML, and UML**. by Timothy J. Grose, Gary C. Doney, Stephen A. Brodsky; Publisher: John Wiley & Sons, ISBN: 0471384291

- **Pràctica tutoritzada.** Enginyeria del Programari de components i sistemes distribuïts, per Josep Maria Camps i Tiba, Universitat Oberta de Catalunya

11. Llista de referències

- http://www.omg.org/technology/documents/formal/uml.htm OMG Unified Modeling Language Specification, v1.3

- http://www.omg.org/technology/documents/formal/xmi.htm MOF 2.0/XMI Mapping Specification, v2.1

- http://www.dcc.uchile.cl/~psalinas/uml Tutorial UML

12. Annexes

12.1. Annex A. Codi generat "CategoriaBean.java". Generat amb MagicDraw

Fitxer situat a *C:\jboveb_producte_MagicDraw\src\cat\uoc\pfc\ejb\CategoriaBean.java*

```
/**
 * @(#) CategoriaBean.java
 */

package src.cat.uoc.pfc.ejb;

import javax.ejb.EntityBean;
import javax.ejb.CreateException;

public abstract class CategoriaBean implements EntityBean
{
        protected EntityContext ctx;
        public ProducteBean 1...*;

        public void categoriaBean( )        {
        }

        /**
         * Mètodes abstractes get/set per la persistencia gestionada pel contenidor
         */
        public abstract String getNom( );
        public abstract void setNom( String nom );
        public abstract String getDescripcio( );
        public abstract void setDescripcio( String descripcio );
        public abstract String getCodi( );
        public abstract void setCodi( String codi );

        public void ejbActivate( ) {
        }

        public void ejbRemove( ) {
        }

        public void ejbPassivate( ) {
        }

        public void ejbLoad( )      {
        }

        public void ejbStore( )     {
        }

        public void setEntityContext( EntityContext ctx )     {
        }

        public void unsetEntityContext( )   {
        }

        public void ejbPostCreate( String codi, String nom, String descripcio )    {
```

```
        }

    /**
     * Mètode d'incicialització que correspon al mètode create de les interficies Home.
     *
     * Quan el client crida al mètode create() d'un Home Object el Home Object (el podeu veure com
     * un proxy) crida al corresponent mètode ejbCreate().
     */
    public void ejbCreate( String codi, String nom, String descripcio ) throws CreateException        {
    }
    }
```

12.2. Annex B. Codi generat "CategoriaBean.java". Generat amb ArgoUML

Fitxer situat a *C:\jboveb_producte_ArgoUml\src\cat\uoc\pfc\ejb\CategoriaBean.java*

```
package src.cat.uoc.pfc.ejb;

import src.cat.uoc.pfc.interfaces.CategoriaLocalHome;
import src.cat.uoc.pfc.interfaces.CategoriaHome;
import javax.ejb.EntityBean;
import src.cat.uoc.pfc.interfaces.Categoria;
import src.cat.uoc.pfc.interfaces.CategoriaLocal;
import java.util.Vector;
import javax.ejb.SessionBean;

/**
 *  EJB d'entitat que modela una categoria de la base de dades.
 *
 *  Fa servir CMP
 *
 *  Una categoria conté un codi, un nom i una descripcio com a camps persistents.
 */
public abstract class CategoriaBean implements CategoriaLocalHome, CategoriaHome, CategoriaLocal,
Categoria, SessionBean, EntityBean, CategoriaLocalHome, CategoriaHome, Categoria, CategoriaLocal
{

  protected EntityContext ctx;

    public Vector  myProducteBean;
    public CategoriaLocalHome myCategoriaLocalHome;
    public CategoriaHome myCategoriaHome;

  public void categoriaBean() {
  }

  /**
   *  ---------------------------------------------------------------
   *  Mètodes abstractes get/set per als caps persistents gestionats pel contenidor
   *  ---------------------------------------------------------------
   */
  abstract public String getNom();
  abstract public void setNom(String nom);
  abstract public String getDescripcio();
  abstract public void setDescripcio(String descripcio);
  abstract public String getCodi();
  abstract public void setCodi(String codi);

  /**
   *  ---------------------------------------------------------------
```

```
     *  Mètodes de l'EJB que son obligatoris.  Aquests mètodes són mètodes de callback
     *  sempre crida el contenidor, els clients mai criden directament a aquests mètodes.
     *  ----------------------------------------------------------------
     */
    public void ejbActivate() {
    }

    public void ejbRemove() {
    }

    public void newOperation() {
    }

    public void ejbPassivate() {
    }

    public void ejbLoad() {
    }

    public void ejbStore() {
    }

    public void setEntityContext(EntityContext ctx) {
    }

    public void unsetEntityContext() {
    }

    public void ejbPostCreate(String codi, String nom, String descripcio) {
    }

    /**
     *  Mètode d'incicialització que correspon al mètode create de
     *  les interficies Home.
     *
     *  Quan el client crida al mètode create() d'un Home Object
     *  el Home Object (el podeu veure com un proxy) crida al corresponent
     *  mètode ejbCreate()
     */
    public void ejbCreate(String codi, String nom, String descripcio) {
    }
}
```