

B. Bibliografía

Web AndroMDA

<http://www.andromda.com/>

Web de OMG sobre MDA

<http://www.omg.org/mda/>

Diversos documentos de OMG

MDA Guide - <http://www.omg.org/cgi-bin/doc?omg/03-06-01>

UML 2 Metamodel - <http://www.omg.org/cgi-bin/doc?ptc/2004-10-05>

Apuntes de asignaturas de ingeniería del software

(ES1: Edicions UPC, ES2: <https://raco.fib.upc.es/servlet/lcfib.racons.Portal>)

Software Architecture Documentation

(Felix Bachmann, Len Bass, Jeromy Carriere, Paul Clements, David Garlan, James Ivers, Robert Nord, Reed Little)

<http://www.sei.cmu.edu/pub/documents/00.reports/pdf/00sr004.pdf>

Software Engineering: A Practitioner's Approach, 6/e

(Roger S Pressman, R.S. Pressman and Associates, ISBN: 0072853182)

MDA Explained: Model Driven Architecture - Practice & Promise

(Warmer, Kleppe & Bast Addison-Wesley, ISBN 032119442X)

B. Bibliografía

Web Andriana

http://www.andriana.com

Web de DMG sobre PDA

http://www.dmg.com

Diversos documentos de DMG

http://www.dmg.com/docs/...
http://www.dmg.com/docs/...

Aportes de ingenieros de ingeniería del software

http://www.dmg.com/docs/...

Software Architecture & Documentation

http://www.dmg.com/docs/...
http://www.dmg.com/docs/...

http://www.dmg.com/docs/...

Software Engineering: A Practitioner's Approach, 6th

http://www.dmg.com/docs/...

PDA Explains: PDA's Evolving Architecture - Practice &

Practice

http://www.dmg.com/docs/...

C. Asignación de tratamientos

Consideraciones

Cualquier tratamiento que requiera el uso de una base de datos tendrá como mínimo una dependencia tecnológica media. En el caso de Hybernate la dependencia será alta ya que además se requiere el uso de ese software. En el caso de disparadores y procedimientos almacenados la dependencia tecnológica será alta pues se depende de los lenguajes internos de las bases de datos.

Los tratamientos que necesiten una interfaz de usuario con una complejidad media o alta tendrán como mínimo una dependencia tecnológica media, ya que esto indica que será necesario usar tecnologías externas para la representación de la interfaz.

La siguiente tabla indica el significado de los distintos colores usados en las tablas.

	Tratamiento por defecto
✓	Permite tratamiento
	Impide tratamiento
	No tiene efecto

Leyenda

Responsabilidad de clave

Tratamientos	Preferencias de usuario											
	Dependencia tecnológica			Lenguaje de desarrollo			Base de datos		Complejidad de la interfaz			
	Baja	Media	Alta	C++	Java	.NET	Ninguna	Oracle	PostgreSQL	Baja	Media	Alta
Dictionary												
Hybernate			✓		✓	✓		✓	✓			
SQL		✓	✓					✓	✓			
DBSchema		✓	✓					✓	✓			

Responsabilidad de cardinalidad

Tratamientos	Preferencias de usuario											
	Dependencia tecnológica			Lenguaje de desarrollo			Base de datos		Complejidad de la interfaz			
	Baja	Media	Alta	C++	Java	.NET	Ninguna	Oracle	PostgreSQL	Baja	Media	Alta
InputSizeControl		✓	✓								✓	✓
Hybernate			✓		✓	✓		✓	✓			
CardinalityControl												
SQL		✓	✓					✓	✓			
Trigger			✓					✓	✓			
DBSchema		✓	✓					✓	✓			

Responsabilidad de restricción textual "existsElement"

Tratamientos	Preferencias de usuario											
	Dependencia tecnológica			Lenguaje de desarrollo			Base de datos		Complejidad de la interfaz			
	Baja	Media	Alta	C++	Java	.NET	Ninguna	Oracle	PostgreSQL	Baja	Media	Alta
Dictionary												
Hybernate			✓		✓	✓		✓	✓			
InputFilter		✓	✓									✓
SQL		✓	✓					✓	✓			

Responsabilidad de restricción textual "notEmptyPopulation"

Tratamientos	Preferencias de usuario											
	Dependencia tecnológica			Lenguaje de desarrollo			Base de datos		Complejidad de la interfaz			
	Baja	Media	Alta	C++	Java	.NET	Ninguna	Oracle	PostgreSQL	Baja	Media	Alta
Dictionary												
Hybernate			✓		✓	✓		✓	✓			
SQL		✓	✓					✓	✓			

Responsabilidad de restricción textual "deleteElement"

Tratamientos	Preferencias de usuario											
	Dependencia tecnológica			Lenguaje de desarrollo			Base de datos		Complejidad de la interfaz			
	Baja	Media	Alta	C++	Java	.NET	Ninguna	Oracle	PostgreSQL	Baja	Media	Alta
Dictionary												
Hybernate			✓		✓	✓		✓	✓			
SQL		✓	✓					✓	✓			
StoredProcedure			✓					✓	✓			
OnCascade		✓	✓					✓	✓			

Responsabilidad de restricción textual "insertElement"

Tratamientos	Preferencias de usuario											
	Dependencia tecnológica			Lenguaje de desarrollo			Base de datos		Complejidad de la interfaz			
	Baja	Media	Alta	C++	Java	.NET	Ninguna	Oracle	PostgreSQL	Baja	Media	Alta
Dictionary												
Hybernate			✓		✓	✓		✓	✓			
SQL		✓	✓					✓	✓			
StoredProcedure			✓					✓	✓			

Responsabilidad de restricción textual "listAll"

Tratamientos	Preferencias de usuario											
	Dependencia tecnológica			Lenguaje de desarrollo			Base de datos		Complejidad de la interfaz			
	Baja	Media	Alta	C++	Java	.NET	Ninguna	Oracle	PostgreSQL	Baja	Media	Alta
Dictionary												
Hybernate			✓		✓	✓		✓	✓			
SQL		✓	✓					✓	✓			

Asignación de tratamientos

D. Manual de instalación

Instalación de AndroMDA

La forma más adecuada de instalar *AndroMDA* es seguir el manual de su pagina web ya que es un proceso algo complejo y propenso a cambios. La última versión de dicho manual se encuentra en la siguiente dirección: http://www.andromda.org/index.php?option=com_content&task=category§ionid=11&id=42&Itemid=89

A continuación describo a grandes rasgos los pasos para instalar *AndroMDA*.

1. Instalar *Java* 5.0: <http://java.sun.com/j2se/1.5.0/download.jsp>
2. Instalar *Maven* 2.0: <http://maven.apache.org/>
3. Instalar *Jboss* (Opcional): <http://labs.jboss.com/>
4. Instalar *MySQL* (Opcional): <http://www.mysql.com/>
5. Instalar *Eclipse* (Opcional): <http://www.eclipse.org/>
6. Instalar *MagicDraw* 11: <http://www.magicdraw.com/>
7. Configurar variables de entorno:

Variable	Contenido	Requerida
JAVA_HOME	Directorio de <i>Java</i>	SI
JBOSS_HOME	Directorio de <i>JBoss</i>	SI
M2_HOME	Directorio de <i>Maven</i>	SI
M2_REPO	Repositorio local de <i>Maven</i> ¹	SI
MAVEN_OPTS	Parámetros a pasar cuando ejecutamos <i>Maven</i>	NO
PATH	Hay que añadir: %JAVA_HOME%\bin;%M2_HOME%\bin	SI

8. Crear el repositorio local.

Ejecutar el siguiente comando en un directorio temporal:

```
mvn archetype:create -DgroupId=testapp -DartifactId=testapp
```

9. Instalar *AndroMDA Application plugin*

Descargar el plugin de la siguiente dirección:

<http://team.andromda.org/maven2/org/andromda/maven/plugins/andromdapp-maven-plugin/3.2/andromdapp-maven-plugin-install-3.2.zip>

Descomprimirlo en el directorio del repositorio de *Maven*.

Una vez completados estos pasos ya tenemos instalado *AndroMDA*. En la web de *AndroMDA* se detallan cada uno de los pasos anteriores.

¹ El repositorio es un directorio que contiene todos los componentes usados por *Maven*.

Instalación del cartridge

Para poder usar el *cartridge*, es necesario haber completado la instalación de *AndroMDA* correctamente. Su instalación es sencilla ya que el entorno de trabajo ya es funcional.

1. Descargar el código fuente del *cartridge*.

Disponible en: <http://www.lsi.upc.edu/~gessi/AR3L/>

2. Descomprimir el código fuente del *cartridge* en un directorio temporal.
3. Ejecutar *Maven* en dicho directorio.

En el proceso de compilación del *cartridge* se crea una copia en el repositorio. Una vez está el *cartridge* en el repositorio ya se puede usar.

E. Manual de usuario

Actualmente no existe una forma mecánica de generar proyectos para el cartridge de asignación de responsabilidades a capas. La forma más fácil es partir del proyecto test. (<http://www.lsi.upc.edu/~gessi/AR3L/>)

Un proyecto para este cartridge es sencillo únicamente tiene dos ficheros de configuración. Se han resaltado los elementos que al usuario podría interesarle cambiar.

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.test.test</groupId>
  <artifactId>test-md</artifactId>
  <packaging>pom</packaging>
  <name>Test MDA</name>
  <version>1</version>
  <description>The Test Project.</description>

  <dependencies>
    <dependency>
      <groupId>org.andromda.profiles.uml2</groupId>
      <artifactId>andromda-profile</artifactId>
      <version>3.2</version>
      <type>xml.zip</type>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>org.andromda.cartridges</groupId>
      <artifactId>andromda-arc-cartridge</artifactId>
      <version>3.2</version>
    </dependency>
  </dependencies>

  <build>
    <defaultGoal>install</defaultGoal>
    <sourceDirectory>
      ${project.basedir}/src/uml
    </sourceDirectory>
    <plugins>
```



```

        <plugin>
            <groupId>org.andromda.maven.plugins</groupId>
            <artifactId>andromda-maven-plugin</artifactId>
            <version>3.2</version>
            <executions>
                <execution>
                    <goals>
                        <goal>run</goal>
                    </goals>
                </execution>
            </executions>
            <configuration>
                <configurationUri>
                    ${pom.basedir}/src/config/andromda.xml
                </configurationUri>
            </configuration>
        </plugin>
    </plugins>
</build>
<properties>
    <model.uri>
        file:${project.build.sourceDirectory}/test.uml2
    </model.uri>
    <filter></filter>
    <validation>true</validation>
    <conf.dir>${pom.basedir}/src/config</conf.dir>
    <common.generated.dir>
        ${pom.basedir}/target
    </common.generated.dir>
</properties>
</project>

```

Código 1: Configuración de Maven "pom.xml"

```

<andromda>
    <properties>
        <property name="modelValidation">${validation}</property>
        <property name="cartridgeFilter">${filter}</property>
    </properties>
    <server>
        <host>localhost</host>
        <port>4447</port>
    </server>
    <repositories>
        <repository name="emf-uml2">
            <models>
                <model type="emf-uml2">
                    <uri>${model.uri}</uri>
                </model>
            </models>
        </repository>
    </repositories>
    <namespaces>
        <namespace name="default">

```



```
        <properties>
            <property name="languageMappingsUri">
                Java
            </property>
            <property name="wrapperMappingsUri">
                JavaWrapper
            </property>
        </properties>
    </namespace>
    <namespace name="arc">
        <properties>
            <property name="responsibilities">
                ${common.generated.dir}
            </property>
            <property name="tecnologicDependency">
                High
            </property>
            <property name="developLanguage">
                .NET
            </property>
            <property name="database">
                PostgreSQL
            </property>
            <property name="interfaceComplexity">
                High
            </property>
        </properties>
    </namespace>
</namespaces>
</andromda>
```

Código 2: Configuración de AndroMDA "src/config/andromda.xml"

Una vez estén modificados al gusto del usuario, sólo queda diseñar el diagrama de clases *UML* usando *MagicDraw* y finalmente ejecutar Maven en el directorio del proyecto para generar la lista de responsabilidades y sus tratamientos.

