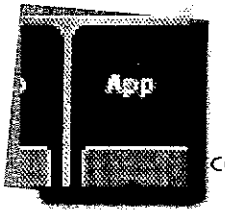
CUORE

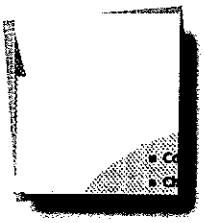
Octubre 2008
Nº 39
6,50€

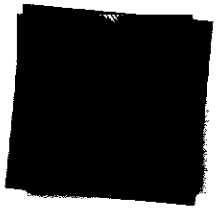
Círculo de Usuarios Oracle de España

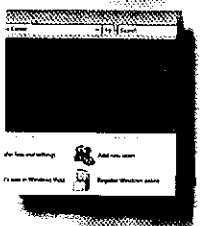
En este número...

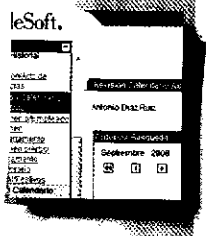
Sumario

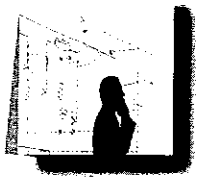


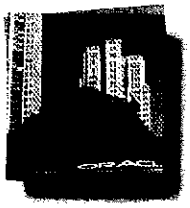


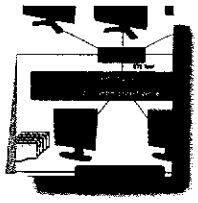


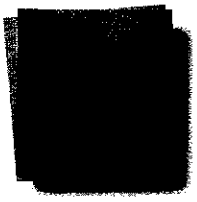




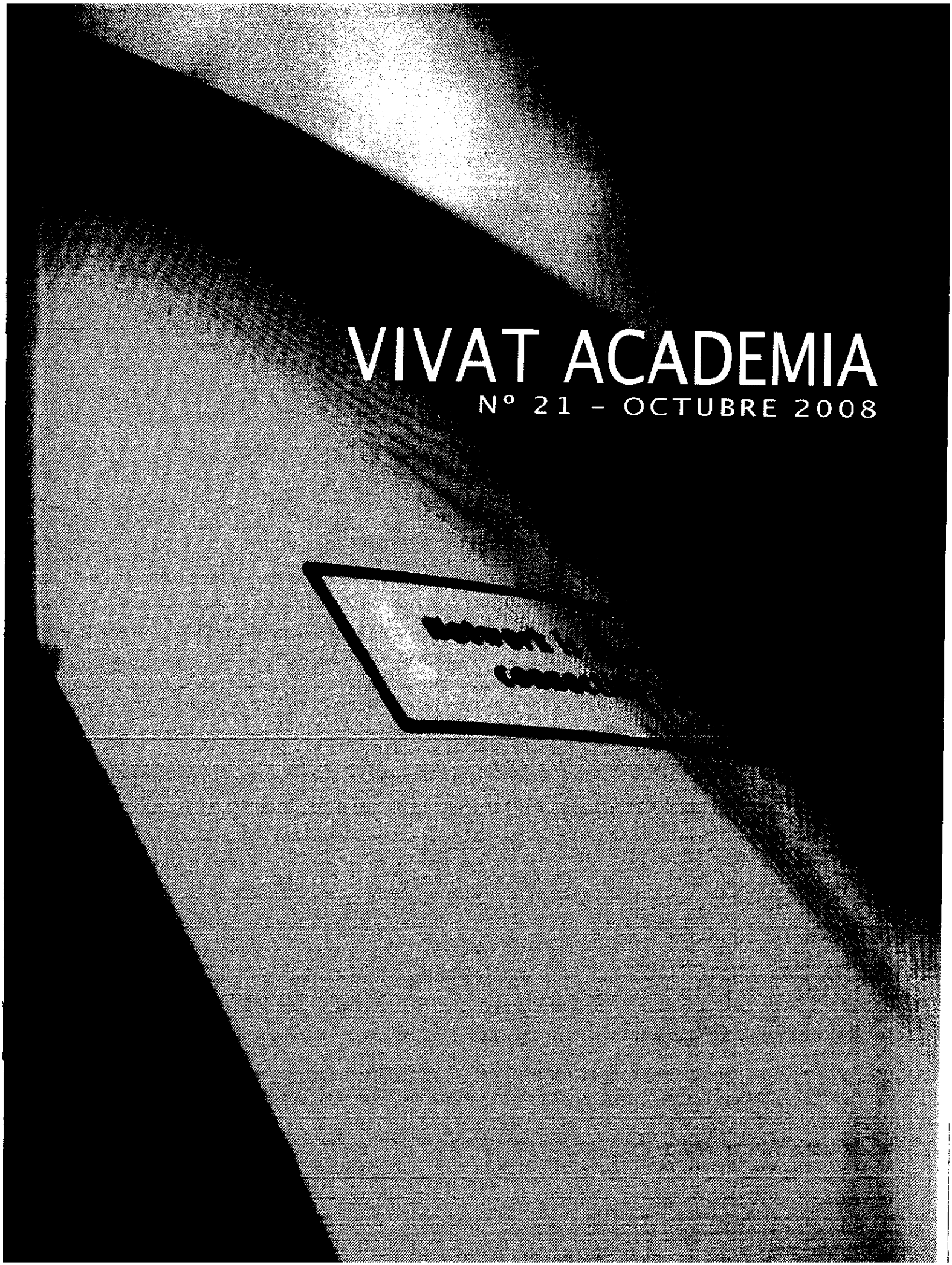








Edita: Cuore. www.cuore.es
Editora: Carmen Gútiez
Consejo Editorial: José Ángel Alonso, Máximo Aborruza, Carmen Gútiez, Rafael Rojo, David Abreu, Patricia Martinet, Marta Eguskiza y José Manuel López.
Diseño y realización: www.nicpublicidad.com



VIVAT ACADEMIA

Nº 21 – OCTUBRE 2008

IMPLEMENTACIÓN DE ALMACENES DE DATOS SEGUROS EN ORACLE LABEL SECURITY

Carlos Blanco[1], Eduardo Fernández-Medina[1], Juan Trujillo[2] y Mario Piattini[1]
Dep. de Tecnologías y Sistemas de Información. Escuela Superior de Informática
Grupo ALARCOS – Instituto de Tecnologías y Sistemas de Información
Universidad de Castilla-La Mancha. Ciudad Real. España
{Carlos.Blanco, Eduardo.Fdezmedina, Mario.Piattini}@uclm.es
[2]Dep. de Lenguajes y Sistemas de Información. Facultad de Informática
Grupo de Investigación LUCENTIA. Universidad de Alicante. Alicante. España
jtrujillo@dlsi.ua.es

RESUMEN

En este artículo se señala la importancia de considerar la definición de medidas de seguridad desde etapas tempranas del desarrollo de sistemas de información, y en particular, en el desarrollo de Almacenes de Datos. Estos almacenes contienen y gestionan información crucial de negocio, que es utilizada para la toma de decisiones y cuyo acceso ha de controlarse, dependiendo de la sensibilidad de los datos, y de los permisos o perfiles de los usuarios que intentan los accesos. Para cumplir ese objetivo, se presenta una estrategia de desarrollo dirigida por modelos para el diseño de Almacenes de Datos, que permite al diseñador considerar aspectos de seguridad en todas las etapas del diseño, y cuya implementación puede ser realizada de forma semi-automática, a partir del diseño lógico detallado. De este modo, este artículo presenta una visión general de este enfoque, y se centra en la implementación final en Sistemas Gestores de Bases de Datos (SGBD) utilizando Oracle Label Security. El artículo trata de evitar los aspectos más teóricos de la propuesta, y está

guiado por un ejemplo sencillo a través de las distintas fases del desarrollo.

1. INTRODUCCIÓN

Es importante comenzar señalando la importancia de la seguridad de la información en la supervivencia de las empresas. La seguridad es un aspecto que ha de ser considerado, no como una medida de interés que es añadida en las últimas etapas del proceso de desarrollo, sino como un aspecto crítico que hà de estar presente desde las primeras etapas como un requisito de vital importancia (Mouratidis y Giorgini 2006). Si la seguridad no se considera desde el principio del desarrollo, los diseños no serán robustos, al no integrar parte de los requisitos de los sistemas. Este hecho provoca la necesidad de integrar las soluciones de seguridad en forma de parches del sistema, y por lo tanto, supone tanto una menor calidad del producto construido, como un mayor esfuerzo de mantenimiento.

En concreto, los Almacenes de Datos son sistemas que manejan un gran volumen de información histórica usada para el proceso de toma de decisiones. Dicha información proviene de fuentes heterogéneas de datos y es organizada de acuerdo a un modelo multidimensional en base a hechos y dimensiones (Kimball 2002). Por lo tanto, incluso con mayor necesidad que en otros tipos de sistemas de información, los Almacenes de Datos requieren que los requisitos de seguridad (especialmente aquellos de confidencialidad en el acceso) estén bien presentes, para evitar que usuarios no autorizados puedan recuperar información sensible para la empresa así como, en algunos casos, información de carácter personal como raza, religión, ideología, enfermedades, etc. que debe estar protegida según las leyes de protección de datos (LOPD).

Por otro lado, el desarrollo de software dirigido por modelos nos permite la construcción del sistema en base a la definición de modelos a varios niveles de abstracción, separando la especificación de la funcionalidad del sistema de aspectos cercanos a la implementación y específicos de la plataforma. Además, el uso de este enfoque soporta la generación automática de código mediante la definición de transformaciones automáticas entre modelos. La OMG propone Model-Driven Architecture, MDA (OMG 2003) como un enfoque dirigido por modelos para el desarrollo de sistemas.

Hay varias iniciativas que han aparecido en el intento de desarrollar Almacenes de Datos seguros. Unas de ellas están centradas en las herramientas finales, como por ejemplo la propuesta de Priebe y Pernul (Priebe y Pernul 2001) en la que desarrollan una metodología de diseño con la que analizan requisitos de seguridad y finalmente los implementan en SQL Server, proponiendo una extensión de su lenguaje de restricciones. Y por otro lado, otras propuestas se centran en la integración de seguridad en el modelado utilizando UML, como "UMLSec" (Jürjens 2004), o en el uso del desarrollo dirigido por modelos, como "Model Driven Security" (Basin et al. 2003), aunque no están diseñadas para su aplicación directa a Almacenes de Datos.

Nuestra propuesta (Fernández-Medina et al. 2007) utiliza el estándar de la OMG para el desarrollo dirigido por modelos, MDA (OMG 2003), y propone una arquitectura de desarrollo de Almacenes de Datos seguros formada por varios modelos a lo largo del proceso de desarrollo, y que permite especificar ciertos aspectos de confidencialidad y auditoría, y que mediante un conjunto de transformaciones (expresadas formalmente) entre modelos, se facilita la generación semi-automática de código que implementa los almacenes de datos seguros.

En las siguientes secciones se muestra un resumen de la arquitectura para el desarrollo de Almacenes de Datos seguros, utilizando un ejemplo y mostrando la implementación final de dichos aspectos de seguridad utilizando Oracle Label Security.

2. ARQUITECTURA PARA EL DESARROLLO DE ALMACENES DE DATOS SEGUROS

En esta sección describimos brevemente nuestra propuesta de arquitectura MDA para el desarrollo de Almacenes de Datos seguros (Fernández-Medina et al. 2007). Este enfoque nos permite modelar el Almacén a distintos niveles de abstracción a la vez que se considera la especificación de medidas de seguridad desde etapas tempranas del proceso de desarrollo. Para ello, nos basamos en un modelo de control de

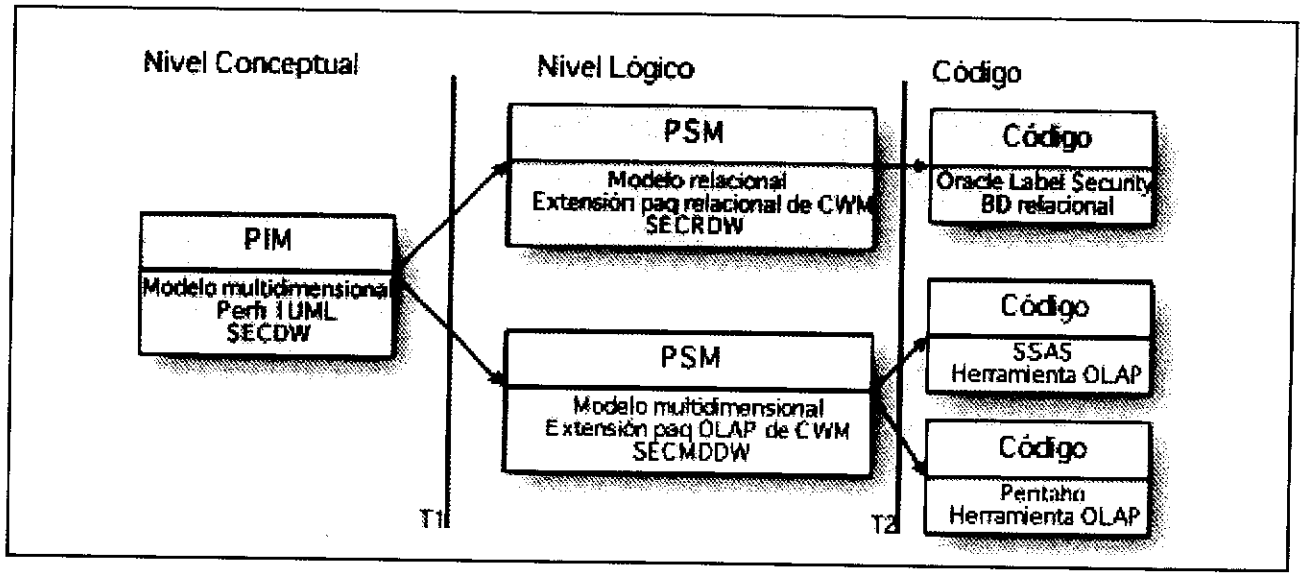


Figura 1. Arquitectura para el Desarrollo de Almacenes de Datos Seguros

acceso y auditoría (ACA) específicamente creado para Almacenes de Datos (Fernández-Medina et al. 2006) que nos permite clasificar la información del sistema y definir reglas de seguridad. La Figura 1 muestra una vista general de esta arquitectura.

Podemos observar cómo en nuestra propuesta existen varios modelos definidos a diferentes niveles de abstracción mediante la extensión de estándares con aspectos de seguridad. A nivel conceptual tenemos un modelo independiente de la plataforma (PIM) y a nivel lógico varios modelos específicos de la plataforma (PSM), uno bajo un enfoque relacional y otro bajo un enfoque multidimensional que nos permiten ir hacia código para Sistemas Gestores de Bases de Datos (SGBD) o herramientas de Procesamiento Analítico en Línea (OLAP) respectivamente. Además, para soportar el paso automático entre modelos desde un nivel alto de abstracción hasta desembocar en la implementación final, han sido definidas reglas de transformación mediante el uso del estándar propuesto por la OMG: Query / Views / Transformations (QVT) (OMG 2005).

2.1. MODELADO CONCEPTUAL

El modelo conceptual, o también conocido como modelo multidimensional en terminología de Almacenes de Datos, realiza el diseño conceptual del Almacén de Datos desde un enfoque independiente de la plataforma. Para

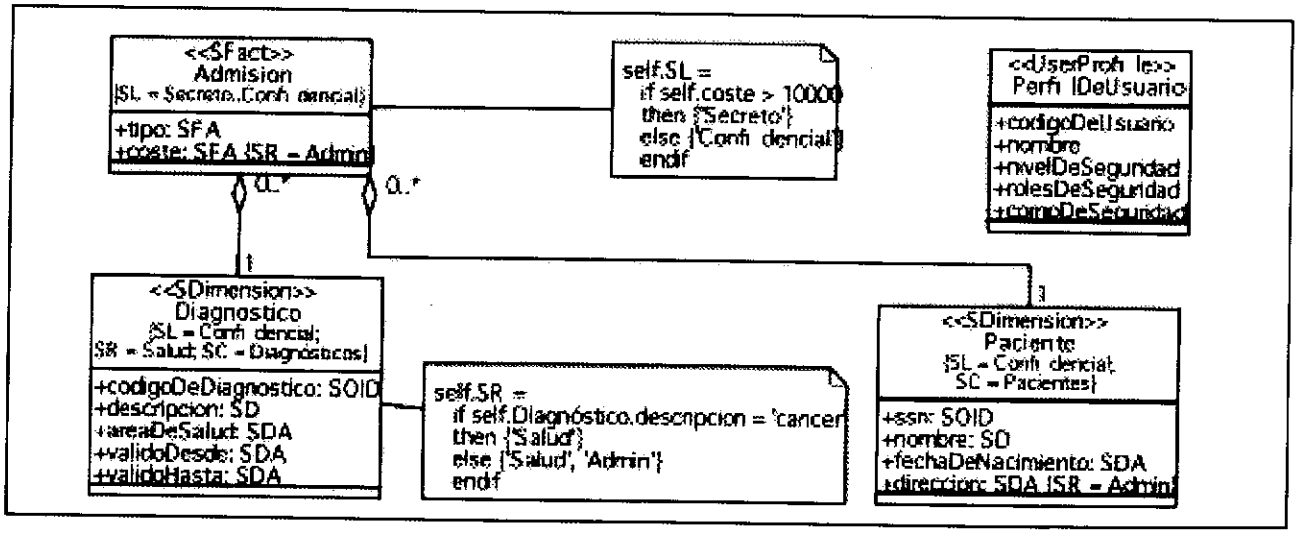


Figura 2. Modelado a nivel conceptual

ello se ha definido un metamodelo multidimensional seguro utilizando UML, llamado SECDW (Fernández-Medina et al. 2007), que soporta las características propias de los Almacenes de Datos (dimensiones, clases bases, relaciones muchos a muchos, etc.) y la inclusión de aspectos de seguridad en base a un modelo de control de acceso y auditoría (ACA) (Fernández-Medina et al. 2006)

Mediante este metamodelo se puede definir conceptualmente la parte estructural del Almacén de Datos usando varios tipos de clases que representan los hechos, dimensiones y clases base presentes en el modelo multidimensional. En cuanto a los aspectos de seguridad, se permite una triple clasificación de los sujetos y objetos del sistema, usando: *niveles* de seguridad que indican el nivel de autorización del usuario; *roles* que organizan a los usuarios en una estructura de roles jerárquica de acuerdo a las responsabilidades de cada tipo de trabajo; y *compartimentos* de seguridad que realizan una clasificación horizontal de los usuarios en compartimentos o grupos. Además, con este metamodelo, se permite la definición de restricciones de seguridad mediante la asignación de reglas de seguridad que actúan sobre las clases y propiedades del modelo.

El modelado multidimensional de Almacenes de Datos organiza los datos de forma orientada a la consulta y el análisis, usando para ello clases hecho, dimensiones y base. De este modo, en el modelado multidimensional las clases hecho representan cubos con las variables a analizar, los cuales tienen asociados dimensiones que organizan los datos por temas, permitiendo así realizar un análisis multidimensional.

La Figura 2 (Pág. V) muestra un ejemplo de modelo conceptual en el que se representa el sistema de admisión de un hospital con varias medidas de seguridad asociadas. En este ejemplo se analizan el tipo y coste de las admisiones, utilizando para ello una clase hecho "Admisión" con esas dos propiedades. Dicha clase se asocia con dos dimensiones, "Paciente" y "Diagnóstico" que han sido representadas por dos clases del tipo "Dimensión" y que permiten la organización de los datos de las admisiones desde dos puntos de vista e incluyen propiedades con datos sobre los pacientes (como nombre, dirección y fecha de nacimiento) y los diagnósticos (como descripción, área de salud o validez).

Para clasificar los sujetos y objetos del sistema se han usado niveles, roles y compartimentos de seguridad. La clasificación de niveles de seguridad consta de los niveles "Secreto", "Confidencial" y "Sin clasificar". Los usuarios del sistema pertenecen al rol "Empleado de Hospital" que se especializa en dos roles, "Salud" y "Administrativo". Y finalmente, los compartimentos de seguridad utilizados son "Pacientes" y "Diagnósticos". Para representar el perfil de seguridad de cada usuario del sistema se ha creado la clase "PerfilDeUsuario" que contiene el nivel, roles y compartimentos asignados al usuario.

Además, en este caso se han especificado sobre el ejemplo varias reglas de seguridad, que pretenden definir con detalle las propiedades que han de cumplir los usuarios para acceder a cierta información sensible, almacenada en nuestro almacén de datos. Esas reglas de seguridad, que en el modelo aparecen como notas asociadas a la clase de hechos y a clases de dimensión (y que contienen especificaciones formales con OCL), tienen la siguiente interpretación:

- La clase hecho "Admisión" sólo puede ser consultada por usuarios con nivel de seguridad "Secreto" si el "coste" de la admisión es mayor a 10000 o por usuarios con nivel "Confidencial" si el "coste" es igual o menor. Además, para consultar el atributo "coste" el usuario ha de tener el rol de "Administrativo".

- La clase dimensión "Diagnóstico" puede ser consultada por usuarios con nivel de seguridad "Confidencial" o mayor, roles "Salud", y que pertenezcan al compartimento de seguridad "Diagnósticos", en el caso en el que la descripción del diagnóstico sea "cáncer", en cualquier otro caso también podría ser consultada por usuarios con el rol "Administrativo".

- La clase dimensión "Paciente" es accesible a usuarios con nivel de seguridad "Confidencial" o mayor, y pertenecientes al compartimento de seguridad "Pacientes". Además, su atributo "dirección" sólo puede ser consultado por usuarios con el rol "Administrativo".

2.2. Modelado Lógico

A nivel lógico y considerando un enfoque relacional, ha sido definida una extensión del paquete relacional del estándar Common Warehouse Metamodel (CWM) llamada SECRDW (Soler et al. 2008), que permite la especificación del Almacén y de sus medidas de seguridad asociadas.

En este caso las clases de hechos, dimensiones y bases definidas a nivel conceptual mediante el modelado multidimensional, pasan a representarse en un modelo específico de la plataforma más cercano a la implementación final, mediante la definición de tablas, relaciones entre tablas y demás elementos como el uso de claves primarias y ajenas. Del mismo modo, los aspectos de seguridad definidos sobre los hechos, dimensiones y bases del modelo multidimensional, pasan a definirse sobre las tablas y atributos correspondientes.

La Figura 3 muestra el modelo lógico correspondiente al ejemplo anterior, representado mediante un esquema lógico en estrella en el que se crean las tablas necesarias, se definen sus atributos, se restablecen las relaciones creando las claves necesarias y se asignan las restricciones de seguridad definidas.

2.3. Implementación en Oracle Label Security

Una vez realizado el modelado multidimensional del Almacén a nivel conceptual y su correspondiente modelo lógico bajo un enfoque relacional, tratamos con su implementación final utilizando para ello Oracle Label Security 11g (Jeloka 2007), la cual puede ser obtenida fácilmente de manera automática partiendo de la especificación realizada a nivel lógico.

Oracle 11g soporta la definición de medidas de seguridad y auditoría mediante el uso de Oracle Label Security (OLS) y Virtual Private Databases (VPD). Mediante OLS podemos definir políticas de seguridad basadas en el uso de etiquetas que especifican información de autorización para los usuarios basándose en los valores de filas o columnas que son insertados o actualizados. Este mecanismo es conocido como función de etiquetado, y con él se definen las políticas de seguridad que son aplicadas a las tablas y atributos de nuestro sistema con el fin de garan-

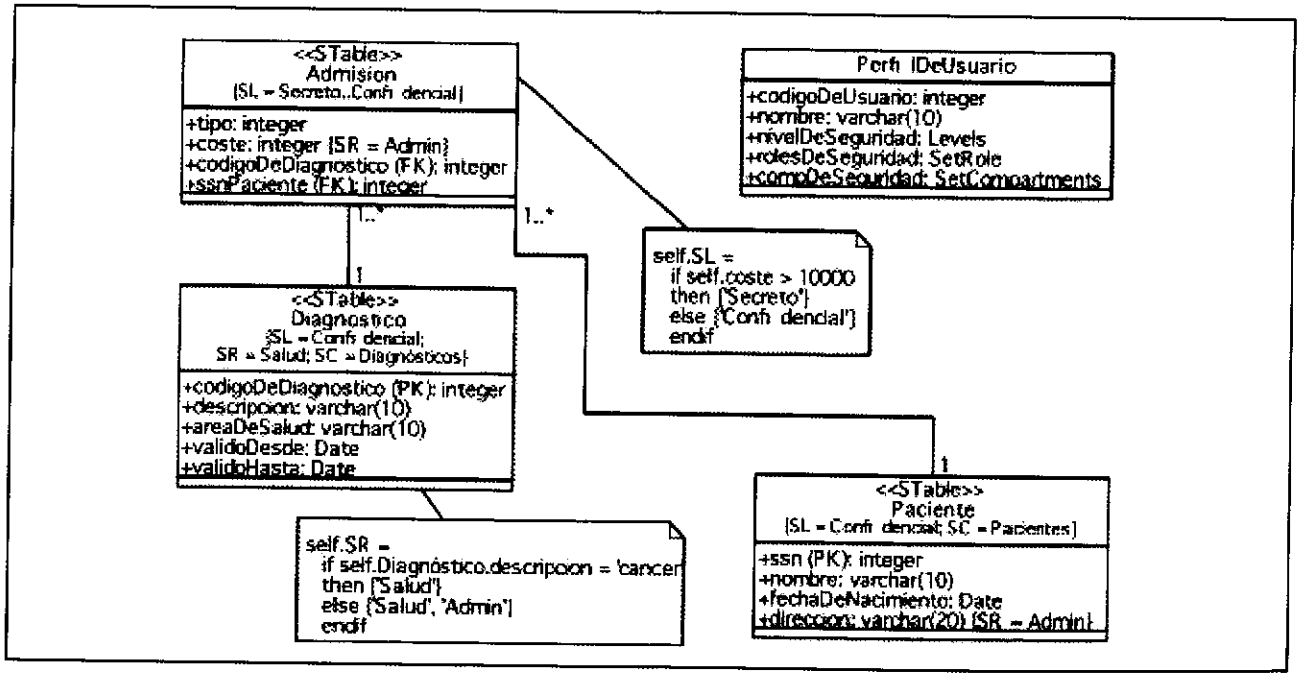


Figura 3: Modelado a nivel lógico.

tizar la confidencialidad de la información. Por otro lado VPD permite restringir el acceso a filas específicas de la tabla, y puede ser usado para definir políticas que permitan a un determinado usuario acceder a un conjunto de datos completamente diferente del que pueden acceder otros usuarios.

En este caso, la implementación de los detalles de seguridad que hayan sido especificadas en el modelado conceptual y lógico del almacén de datos, será relativamente directo, ya que afortunadamente los principales conceptos de seguridad abordados (roles, niveles y compartimentos), así como las restricciones especificadas, son soportados casi directamente por OLS. Sin embargo, para la implementación de almacenes de datos seguros en otras herramientas, hay que realizar procesos de generación de código mucho más complejos. A continuación se muestra la forma de implementar los detalles de seguridad que han sido definidos en el ejemplo. Para ello: (1) se ha de definir una política de seguridad; (2) se ha de establecer la configuración de seguridad del sistema creando los niveles, roles y compartimentos de seguridad necesarios; (3) se han de definir funciones que según las condiciones establecidas en las reglas de seguridad que queremos considerar, establezcan etiquetas con los niveles, roles y compartimentos que pueden acceder a los datos en cada caso; y (4) se ha de aplicar la política creada a las tablas del Almacén.

En primer lugar, tal y como podemos ver en la Tabla 1, se crea la política de seguridad a la que le asignamos el nombre "MiPolitica" y una columna para las etiquetas "MiEtiqueta" que va estar oculta para los usuarios. Además, mediante la modificación de los privilegios de los usuarios establecemos en la política que éstos sólo van a poder tener privilegios de lectura, ya que los Almacenes de Datos no son modificados por los usuarios.

A continuación, se establece la configuración de seguridad del sistema en base a niveles, roles y compartimentos de seguridad (Tabla 1). Para especificar el conjunto de niveles de seguridad utilizados, "Secreto", "Confidencial" y "Sin especificar", se crea cada uno de ellos individualmente mediante "CREATE_LEVEL" y la asignación de un identificador y un nombre corto y largo para el nivel. Posteriormente, utilizando el procedimiento "SET_LEVELS" se define el nivel máximo ('Secreto'), el mínimo ('Sin especificar'), el nivel por defecto ('Confidencial') y el nivel aplicado a las filas ('Confidencial').

El conjunto de roles se define de forma análoga, creando en primer lugar cada rol de forma individual mediante "CREATE_GROUP", salvo que en este caso, el último parámetro del procedimiento

CREATE_POLICY('MiPolitica', 'MiEtiqueta', 'HIDE'); SET_USER_PRIVS('MiPolitica', 'Usuario1', 'READ');
CREATE_LEVEL('MiPolitica', 1, 'LS', 'Secreto'); CREATE_LEVEL('MiPolitica', 2, 'LC', 'Confidencial'); CREATE_LEVEL('MiPolitica', 3, 'LSE', 'Sin especificar'); SET_LEVELS('MiPolitica, 'Usuario1', 'LS', 'LSE', 'LC', 'LC');
CREATE_GROUP('MiPolitica', 1, 'GEH', 'EmpleadoDeHospital', NULL); CREATE_GROUP('MiPolitica', 2, 'GS', 'Salud', 'GEH'); CREATE_GROUP('MiPolitica', 3, 'GA', 'Administrativo', 'GEH'); SET_GROUPS('MiPolitica', 'Usuario1', 'GEH, GS, GA', NULL, 'GEH, GS, GA', 'GEH, GS, GA');
CREATE_COMPARTMENT('MiPolitica', 1, 'CP', 'Pacientes'); CREATE_COMPARTMENT('MiPolitica', 2, 'CD', 'Diagnósticos'); SET_COMPARTMENTS('MiPolitica', 'Usuario1', 'CP, CD', NULL, 'CP, CD');

Tabla 1: Configuración de seguridad del sistema

establece el padre del rol dentro de la jerarquía de roles que queremos definir, en este caso un rol "Empleado de Hospital" que tiene dos roles hijos "Salud" y "Administrativo". Mediante "SET_GROUPS" establecemos el conjunto de roles permitido para lectura, usado por defecto y usado a nivel de fila, estableciendo a NULL la lista de roles permitidos para la escritura.

Del mismo modo, mediante "CREATE_COMPARTMENT" son creados los compartimentos de seguridad "Pacientes" y "Diagnósticos" y mediante SET_COMPARTMENTS" se define el conjunto de compartimentos permitidos para la lectura, por defecto, para escritura y usados a nivel de fila.

Una vez creada la política y la configuración de seguridad pasamos a crear las funciones necesarias para establecer los valores de las etiquetas según las reglas de seguridad definidas, y aplicar la política de seguridad a las tablas correspondientes (ver Tablas 2, 3 y 4, Págs. IX y X).

De este modo podemos ver cómo en la Tabla 2 se define una función de etiquetado para representar las restricciones de la tabla "Admisión", en la que dependiendo de si el valor del atributo "coste" es mayor o menor o igual que 10000 se le asignan unos valores u otros indicando el nivel, roles y compartimentos de seguridad requeridos para acceder a los datos. En este caso el nivel de seguridad exigido varía de "Secreto" a "Confidencial" dependiendo de la condición. Finalmente, la política creada es aplicada a la tabla "Admisión".

A continuación, en la Tabla 3 se define una función de etiquetado para representar las reglas de seguridad definidas sobre la tabla "Diagnóstico". En ella, dependiendo de si el diagnóstico es 'cancer' o no, se les permite la lectura o sólo a los usuarios con el rol "Salud" o también a los usuarios con el rol "Administrativo". Tras definir la etiqueta, se aplica la política a la tabla "Diagnostico".

Finalmente, en la Tabla 4 se crea una función de etiquetado para permitir el acceso a la tabla "Paciente" en la que se establece el perfil de seguridad (nivel, rol y compartimento de seguridad) que han de tener los usuarios para permitirles el acceso. En este caso, según las restricciones de seguridad establecidas el nivel de seguridad debe ser "Confidencial", el rol puede ser "Salud" o "Admin" y compartimento de seguridad ha de ser "Pacientes". Tras definir la etiqueta se aplica la política a la tabla "Paciente".

3. CONCLUSIONES Y TRABAJO FUTURO.

La seguridad de la información es un aspecto de vital importancia que debe ser considerado como un requisito desde las etapas más tempranas del desarrollo de sistemas de información. Particularmente hemos visto cómo los Almacenes de Datos manejan una gran cantidad de datos sensibles, tanto desde el punto de vista de la empresa como desde el de la protección de información de carácter personal, que han de ser protegidos definiendo restricciones de seguridad.

Por otro lado ha sido presentada una arquitectura para el desarrollo de Almacenes de Datos seguros, que utilizando el enfoque de desarrollo dirigido por modelos define el Almacén a varios niveles de abstracción utilizando modelos seguros que añaden aspectos de seguridad en todas las etapas del proceso de desarrollo. El uso de este enfoque también facilita la generación automática de código final mediante la

```
CREATE FUNCTION FuncionAdmision1 (coste: integer) Return LBSCSYS.LABC_LABEL
As MiEtiqueta varchar2(80);
Begin
  If coste > 10000 then
    MiEtiqueta := 'Secreto::EmpleadoDeHospital::Pacientes,Diagnosticos';
  Else
    MiEtiqueta := 'Confidencial::EmpleadoDeHospital::Pacientes,Diagnosticos';
  Endif;
  Return TO_LBAC_DATA_LABEL ('MiPolitica', 'MiEtiqueta');
End;

APPLY_TABLE_POLICY ('MiPolitica', 'Admision', 'Scheme', 'FuncionAdmision1');
```

Tabla 2: Función de etiquetado para Admisión

```
CREATE FUNCTION FuncionDiagnostico1 (descripcion: varchar(10)) Return
LBSCSYS.LABC_LABEL
As MiEtiqueta varchar2(80);
Begin
  If descripcion = 'cancer' then
    MiEtiqueta := 'Confidencial::Salud::Diagnosticos';
  Else
    MiEtiqueta := 'Confidencial::Salud,Admin::Diagnosticos';
  Endif;
  Return TO_LBAC_DATA_LABEL ('MiPolitica', 'MiEtiqueta');
End;
APPLY_TABLE_POLICY('MiPolitica', 'Diagnostico', 'Scheme', 'FuncionDiagnostico1');
```

Tabla 3: Función de etiquetado para Diagnóstico

```
CREATE FUNCTION FuncionPaciente1 () Return LBSCSYS.LABC_LABEL
As MiEtiqueta varchar2(80);
Begin
  MiEtiqueta := 'Confidencial::EmpleadoDeHospital::Pacientes';
  Return TO_LBAC_DATA_LABEL ('MiPolitica', 'MiEtiqueta');
End;
APPLY_TABLE_POLICY ('MiPolitica', 'Paciente', 'Scheme', 'FuncionPaciente1');
```

Tabla 4: Función de etiquetado para Paciente

definición de reglas de transformación entre los modelos definidos.

En este artículo, además de ilustrar varias de las etapas del proceso de desarrollo con un ejemplo, se ha tratado con la implementación final de las medidas de seguridad definidas utilizando para ello Oracle Label Security (OLS). Del mismo modo se ha comprobado como OLS da soporte tanto a la clasificación de sujetos y objetos del sistema en base a niveles, roles y compartimentos de seguridad, como a la definición de políticas de seguridad sobre los distintos elementos del Almacén.

Como trabajo futuro, estamos trabajando en la compleción de esta arquitectura. Por un lado incorporando nuevas restricciones de seguridad que debieran ser contempladas en el proceso de desarrollo y por otro lado analizando e implementando los sistemas seguros en otras herramientas con el fin de poder definir reglas de transformación que nos permitan tanto generar código seguro de forma automática como facilitar la ingeniería inversa y el paso de código seguro de unas plataformas a otras.

4. AGRADECIMIENTOS

Esta investigación es parte del proyecto ESFINGE (TIN2006-15175-C05-05) del Ministerio de Educación y Ciencia y de los proyectos MISTICO (PBC-06-0082) y QUASIMODO (PAC08-0157-0668) de la Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha y el FEDER.

5. BIBLIOGRAFÍA

Basin, D., J. Doser, et al. (2003). Model Driven Security for Process-oriented Systems. ACM Symposium on Access Control Models and Technologies, Como, Italy, ACM Press.

Fernández-Medina, E., J. Trujillo, et al. (2007). "Model Driven Multidimensional Modeling of Secure Data Warehouses." European Journal of Information Systems 16: 374-389.

Fernández-Medina, E., J. Trujillo, et al. (2006). "Access Control and Audit Model for the Multidimensional Modeling of Data Warehouses." Decision Support Systems 42: 1270-1289.

Fernández-Medina, E., J. Trujillo, et al. (2007). "Developing Secure Data Warehouses with a UML extension." Information Systems 32(6): 826-856.

Jeloka, S. (2007). "Oracle Label Security Administrator's Guide, 11g."

Jürjens, J. (2004). Secure Systems Development with UML, Springer-Verlag.

Kimball, R. (2002). The Data Warehouse Toolkit 2ª Edition, John Wiley & Sons.

Mouratidis, H. y P. Giorgini (2006). An Introduction. Integrating Security and Software Engineering: Advances and Future Visions, Idea Group Publishing.

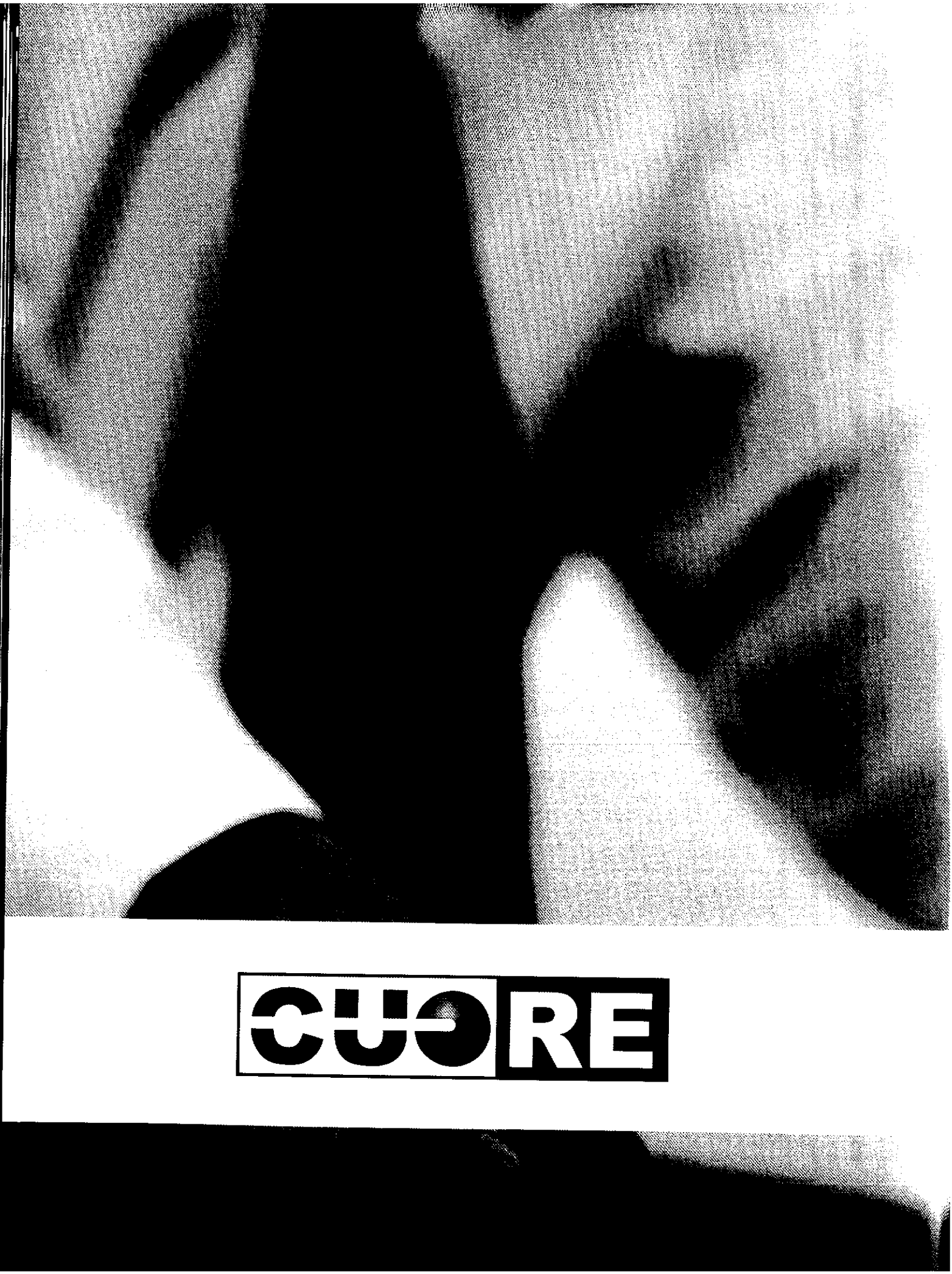
OMG. (2003). "Model Driven Architecture Guide Version 1.0.1." from http://www.omg.org/mda/.

OMG (2005). "MOF QVT final adopted specification."

Priebe, T. y G. Pernul (2001). A Pragmatic Approach to Conceptual Modeling of OLAP Security. 20th International Conference on Conceptual Modeling (ER 2001), Yokohama, Japan, Springer-Verlag.

Soler, E., J. Trujillo, et al. (2008). "Building a secure star schema in data warehouses by an extension of the relational package from CWM." Computer Standard and Interfaces 30(2008): 341-350.

CUORE


CUORE