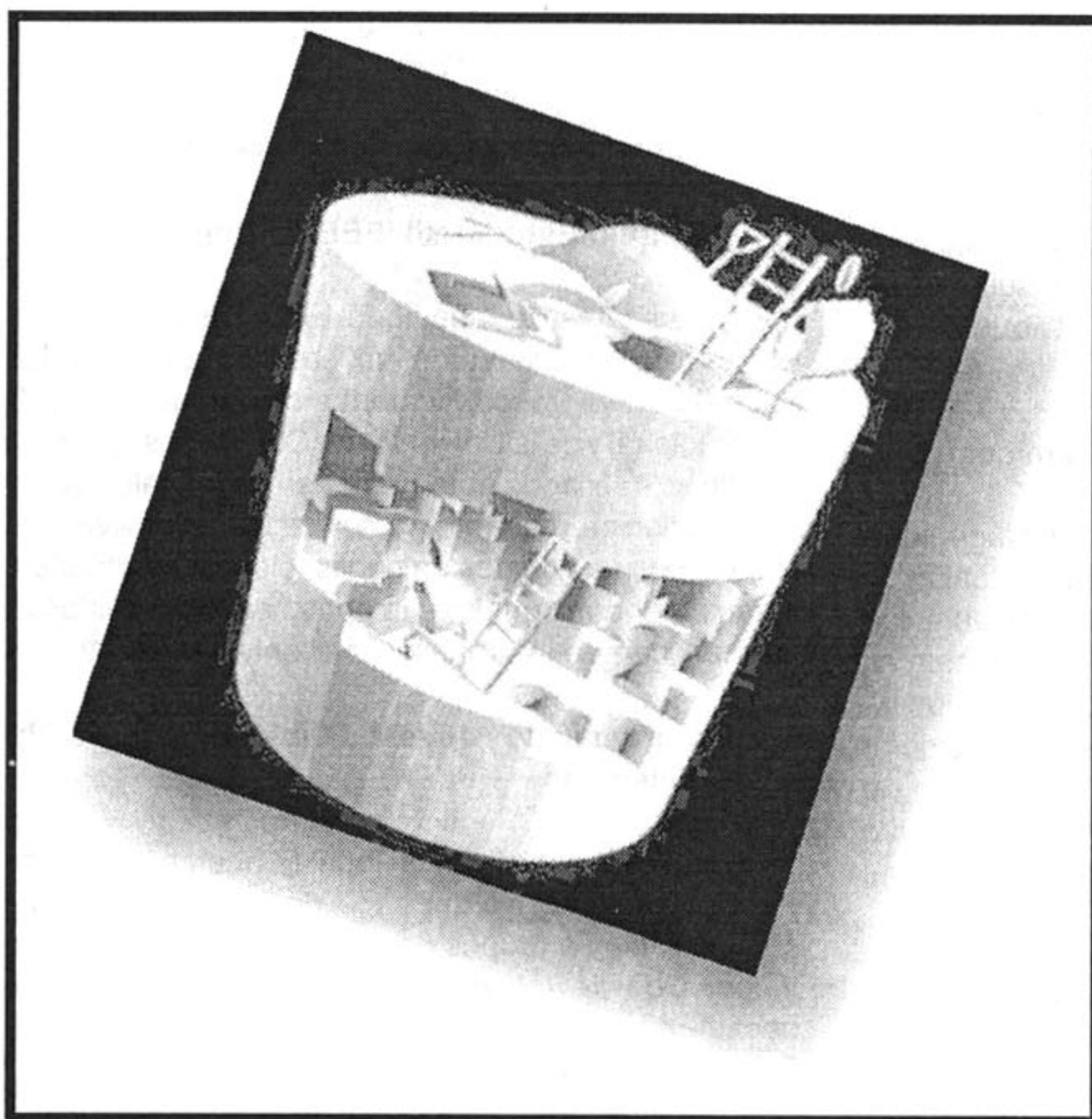


Arquitectura de software para descubrimiento de conocimiento



RESUMEN

Este artículo presenta una propuesta de arquitectura de software para sistemas de descubrimiento de conocimiento, la descripción detallada de cada uno de sus componentes y una aplicación de dicha arquitectura en el desarrollo de un sistema de descubrimiento de conocimiento en datos de prueba de corto circuito para el Laboratorio GRALTA de la Universidad del Valle en Cali, Colombia.

PALABRAS CLAVES

Arquitectura de software, descubrimiento de conocimiento, minería de datos, pruebas de corto circuito.

ABSTRACT

This paper proposes a software architecture for knowledge discovery systems. A detail description for each one of its components and an application for such architecture are described. The architecture was used to develop a knowledge discovery system applied to testing data for a short-circuit in the GRALTA Laboratory at the Universidad del Valle.

KEYWORDS

Software architecture, knowledge discovery, data mining, short circuit tests.

□ **María E. Valencia de Abadía. M.Sc.**
Profesora Titular. EISC.
Universidad del Valle
Cali, Colombia.

□ **Carlos M. Gaona Cuevas. Ph.D.**
Profesor Asociado. EISC.
Universidad del Valle.
Cali, Colombia.

□ **Marta Elena Millán González. Ph.D.**
Profesora Titular. EISC.
Universidad del Valle.
Cali, Colombia.

□ **Jesús A. Aranda Bueno. Ing.**
Profesor Auxiliar. EISC.
Universidad del Valle.
Cali, Colombia.

Grupo GEDI, Universidad del Valle
{maeva, mgaona, millan,
jesarana}@eisc.univalle.edu.co

* La Revista recibió este artículo en Mayo de 2.001

1. INTRODUCCIÓN

La mayoría de las organizaciones cuentan con sistemas de gestión de bases de datos. Sin embargo, normalmente se limitan a registrar, almacenar y consultar los datos relacionados con la operación de la organización durante el tiempo de vida de la misma. Las herramientas de consulta no permiten fácilmente encontrar conocimiento *oculto* en las bases de datos. Si usted sabe exactamente lo que quiere buscar, será suficiente para satisfacer su necesidad, con hacer una consulta sobre una base de datos pero si sólo tiene una idea aproximada de lo que busca, deberá hacer descubrimiento de conocimiento y las técnicas de *data mining* (minería de datos) podrían ayudarle.

Descubrir conocimiento en grandes volúmenes de datos que se encuentran almacenados ya sea en archivos o en bases de datos es un proceso interactivo e iterativo que implica, por una parte, integrar en una herramienta, interrogaciones, análisis y métodos de visualización que le faciliten al usuario entender e interpretar los datos. Para ésto, generalmente es necesario realizar tres grandes etapas: preprocesar los datos, hacer *data mining* y presentar resultados.

El preprocesamiento de los datos puede estar constituido por tareas de limpieza y de transformación. Las tareas de limpieza tienen por objeto eliminar datos erróneos, incompletos o duplicados y las tareas de transformación adecúan la estructura de los datos a través de discretización, jerarquización de atributos y reducción tanto vertical como horizontal. Estas tareas facilitan la etapa siguiente, de minería de datos, a partir de la cual se deriva el conocimiento descubierto. Los resultados de este descubrimiento se despliegan utilizando técnicas de visualización.

El proceso de descubrimiento de conocimiento se puede aplicar en diferentes dominios por ejemplo, para descubrir relaciones implícitas existentes entre síntomas y enfermedades, entre características técnicas y diagnóstico del estado de equipos y máquinas, para determinar perfiles de clientes fraudulentos (evasión de impuestos), para determinar perfiles de estudiantes "académicamente exitosos" en términos de sus características socioeconómicas, entre otros.

Para el desarrollo de los sistemas de descubrimiento de conocimiento es importante considerar las arquitecturas de software aplicables.

Una arquitectura de software es una organización lógica de la forma en que se dispone un conjunto de elementos que van a interactuar para llevar a cabo una tarea. La arquitectura determina la funcionalidad y

modo de interoperación de sus elementos.

En este artículo se propone una arquitectura de software para el desarrollo de sistemas de descubrimiento de conocimiento en grandes volúmenes de datos (archivos o bases de datos). Se describe en detalle cada uno de sus componentes y los procesos asociados. Esto permite tener una conceptualización clara y precisa de los componentes de un sistema de descubrimiento de conocimiento y de las relaciones que se dan entre ellos.

Se presenta también la aplicación de la arquitectura propuesta para desarrollar un sistema de descubrimiento de patrones de comportamiento en bases de datos de pruebas de aptitud al cortocircuito para el Laboratorio de Alta Tensión GRALTA, de la Universidad del Valle.

2. BASES TEÓRICAS

2.1. El Descubrimiento de Conocimiento

El término KDD (Knowledge Discovery in Database) o Descubrimiento de conocimiento en Bases de Datos (DCBD) acuñado en 1989 [12] hace énfasis en que el conocimiento es un producto de un proceso de descubrimiento guiado por los datos. Es un proceso no trivial que busca, a partir de los datos o registros de una base de datos, identificar patrones válidos, novedosos, potencialmente útiles y comprensibles. [5, 6].

Los datos hacen referencia a un conjunto de hechos y un patrón a una expresión útil para describir un subconjunto de los datos o un modelo aplicable a éstos. El término *proceso* implica que DCBD se constituye de varias etapas que se ejecutan iterativa e interactivamente. El proceso es *no trivial* en el sentido en que éste incluye acciones de cierta complejidad que involucran la búsqueda de estructuras, modelos y parámetros en la base de datos. Los patrones que se obtienen deben ser *válidos* con algún grado de certeza, *novedosos* preferiblemente para el usuario al que deberán reportar algún tipo de utilidad.

El proceso DCBD (Figura 1) comienza con la definición y comprensión de un determinado problema y termina con el análisis de los resultados. Una de las propuestas más ampliamente extendida sobre las etapas o fases componentes del proceso DCBD se presenta en [5, 6] e incluye la comprensión del problema, la selección de los datos objetivo, su limpieza y preprocesamiento, la transformación y aplicación del método de descubrimiento o *data mining* a utilizar y la interpretación de los patrones obtenidos o análisis de resultados.

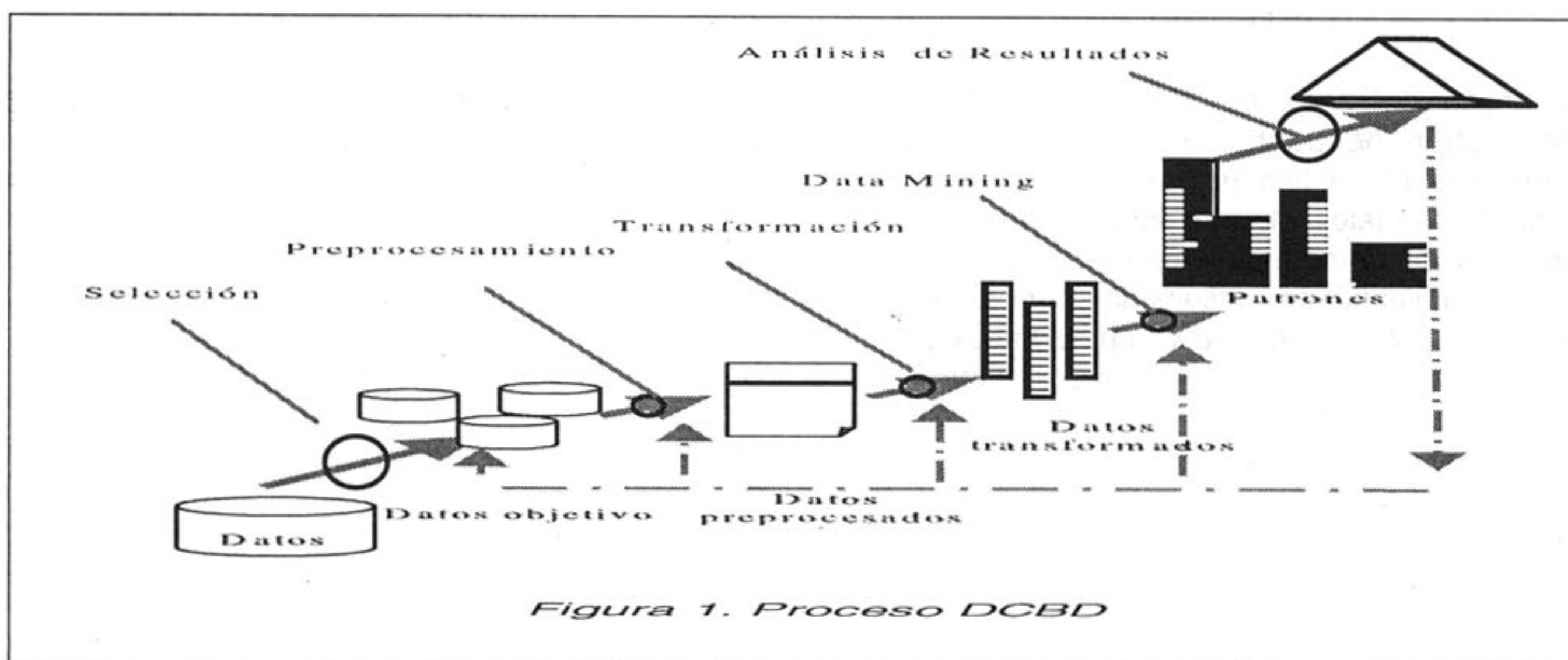


Figura 1. Proceso DCBD

2.2. Las Arquitecturas de Software

Las arquitecturas de software han estado recibiendo mucha atención en los últimos años, dado que desempeñan un rol significativo en el proceso de desarrollo de software. Estas arquitecturas representan la organización estructural en el proceso de construcción sistemática de productos de software.

Las arquitecturas de software se definen como una estructura a alto nivel, que se muestran como una colección de componentes que interactúan, dado que es una representación abstracta que facilita el diseño y la construcción de sistemas complejos y en ella se muestran sus especificaciones y cómo funcionarán en un sistema. En la práctica es una representación lógica de todos los componentes software, cómo se comunican éstos y su disposición organizacional para interactuar y llevar a cabo las tareas definidas para un sistema. [3, 7, 14]

La arquitectura de software que se caracteriza en este artículo hace referencia a la organización estructural de los componentes software a diferencia de las arquitecturas que típicamente se han propuesto para sistemas de descubrimiento de conocimiento (débil, mediana y fuertemente acopladas con Sistemas de Gestión de Bases de Datos) que se refieren, fundamentalmente, a la forma como los sistemas de descubrimiento de conocimiento se integran con Sistemas de Gestión de Bases de Datos [1, 2, 4, 8, 9, 10, 11, 13].

3. ARQUITECTURA DE SOFTWARE PARA DESCUBRIMIENTO DE CONOCIMIENTO

La arquitectura propuesta está enfocada a describir los elementos que se requieren para realizar el proceso de descubrimiento de conocimiento y su integración dentro de un esquema funcional.

La Figura 2 muestra un esquema de la arquitectura propuesta. El usuario interactúa con la aplicación a través de los elementos de interacción que ofrece la *Vista* que es un mecanismo para visualizar los resultados actuales de un proceso de descubrimiento de conocimiento. La *Vista* entrega las entradas del usuario al *Controlador*, con las cuales éste define un conjunto de elementos para especificar un contrato.

El *Contrato* está constituido por una serie de acciones que se deben realizar tales como seleccionar el subconjunto de datos, definir el algoritmo de limpieza a utilizar y especificar los parámetros asociados con ese algoritmo de limpieza, especificar la tarea de data mining a realizar y finalmente, notificar al Gestor de limpieza de Datos.

El *Gestor de Limpieza de Datos* tiene como función disminuir, en alguna medida, el nivel de contaminación de los datos, aumentando de esta manera su calidad. Estos datos son consultados de un *Modelo Inicial de Datos*, que representa los datos originales. El gestor de limpieza debe tratar con distintos tipos de contaminación o ruido: datos faltantes porque no son aplicables o porque no se tenían disponibles al momento de la captura, datos fuera de rango, datos duplicados y datos erróneos y como resultado, produce un *Modelo de Datos Limpios*, que contiene los datos del modelo inicial con menos ruido o contaminación.

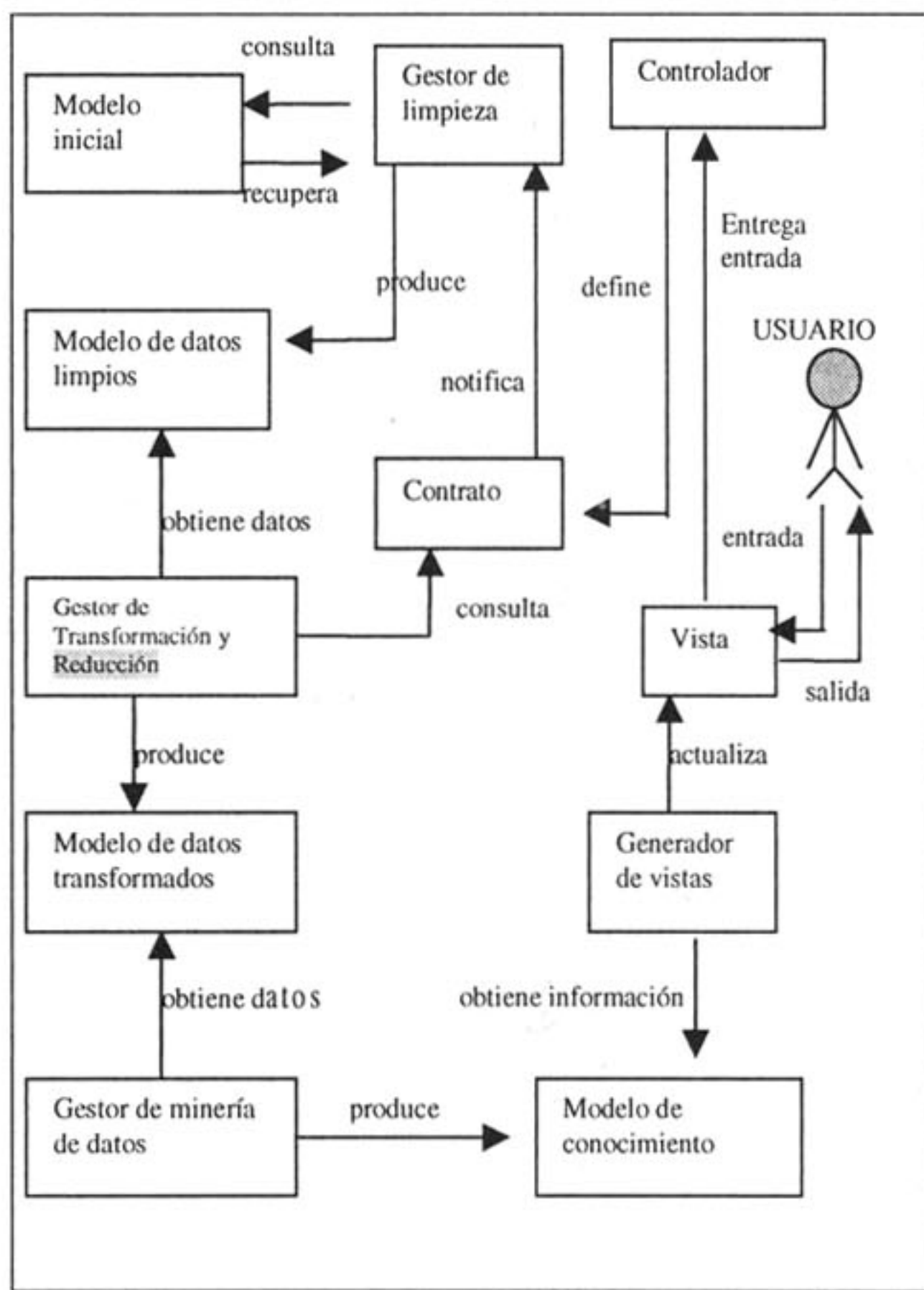


Figura 2. Arquitectura de Software propuesta.

Los datos del Modelo de Datos Limpios son obtenidos por el *Gestor de Transformación y Reducción*. La función primordial de este gestor es adaptar el formato de los datos originales de acuerdo con la tarea de data mining especificada y con el algoritmo a utilizar en cada caso. También es función de él tratar las variables, dependiendo de su tipo, normalizándolas o discretizándolas si fuese necesario. Este gestor produce un nuevo modelo de los datos denominado *Modelo de Datos Transformados* que representa los datos a los cuales se les aplicarán las tareas de data mining.

El *Gestor de Minería de Datos* obtiene datos del *Modelo de Datos Transformados*. Este gestor es uno de los más importantes, dado que en él se llevan a cabo tareas de data mining: clasificación, asociación, «clustering», patrones de secuencia. Su función principal es transformar, de acuerdo con el contrato establecido, el conjunto de datos en reglas o patrones de comportamiento que constituyen el Modelo de Conocimiento.

El *Generador de Vistas* obtiene la información del *Modelo de Conocimiento* y actualiza la *Vista*. El proceso de descubrimiento de conocimiento puede realizarse de manera iterativa de acuerdo a las entradas del usuario.

4. APLICACIÓN DE LA ARQUITECTURA PROPUESTA A UN SISTEMA DE DESCUBRIMIENTO DE CONOCIMIENTO EN DATOS DE CORTO CIRCUITO

Se desarrolló para el laboratorio de alta tensión del grupo GRALTA de la Universidad del Valle el software denominado KDD-PCC para descubrimiento de conocimiento en los datos correspondientes a las pruebas de corto circuito realizadas a los transformadores.

Esta aplicación se implementó en lenguaje JAVA 1.2 y utiliza los paquetes Swing para el desarrollo de la interfaz de usuario, y maneja, para el almacenamiento de la información bases de datos Oracle 8i versión 8.1.6. Se utilizó JDBC para establecer la conectividad entre la base de datos y la aplicación e interfaz gráfica. A continuación se presenta la aplicación realizada, de la arquitectura propuesta a este software y se hace la descripción correspondiente a cada uno de los elementos que la constituyen.

Vista

Se refiere a la interfaz con la cual se comunica el usuario con el sistema y se encarga de visualizar el conjunto de datos obtenido de las pruebas de corto circuito. Permite también al usuario, seleccionar un subconjunto de esos datos, elegir si se hace tratamiento de datos fuera de rango, establecer en cuántos intervalos se discretizan los datos continuos y definir valores de soporte y confianza para el cálculo de reglas de asociación. Al final del proceso de descubrimiento de conocimiento, la vista también despliega al usuario los resultados editados del proceso de obtención de reglas de asociación.

En la Figura 3 se muestra una de las interfaces que permiten al usuario interactuar con el sistema, concretamente estableciendo cuáles atributos se van a tener en cuenta al momento de realizar descubrimiento de conocimiento.

discretización por defecto), o asistida por el usuario (en la cual él define los intervalos de discretización). Este gestor produce el Modelo de datos transformado.

En la Figura 5 se muestra la interfaz a través de la cual el usuario interactúa con el sistema para seleccionar el tipo de técnica de discretización que deberá realizar el gestor de transformación.

Modelo de datos transformados

Contiene la representación de la información que ha sido convertida en valores numéricos enteros consecutivos por el *Gestor de Transformación*.

Gestor de Minería de datos

Se encarga de llevar a cabo la tarea del cálculo de las reglas de asociación, a partir de los datos descritos por el Modelo de datos transformados. Esta tarea ha permitido descubrir patrones informativos como el siguiente: <nucleo = enrollado>, <disposicion devanado = ABA> -> <anormalidad = falla devanado alta tension>, confianza = 66.7%, soporte = 10% cuya interpretación en un lenguaje más natural es dada por el generador de vistas.

Este gestor de minería se ha implementado utilizando el algoritmo A priori mencionado en [1].

En la Figura 6 se muestra la interfaz a través de la cual el usuario establece los parámetros para el funcionamiento del gestor de minería de datos

Modelo de conocimiento

Representa la información de todas las reglas de asociación, con su correspondiente soporte y confianza, obtenidas por el Gestor de Minería de Datos, de los datos de corto circuito descritos por el modelo de datos transformados.

Generador de Vistas

Se encarga de llevar a un lenguaje más natural las reglas de asociación descubiertas para que el usuario tenga mayor claridad acerca del conocimiento descubierto. Ejemplo: para la regla de asociación enunciada en el gestor de minería, la interpretación que se da es la siguiente: el 10% de todos los registros de los transformadores tiene nucleo enrollado, disposición de devanado ABA y presentan anomalía en el devanado de alta tensión. El 66.7% de los transformadores que tienen núcleo enrollado y disposición de devanado

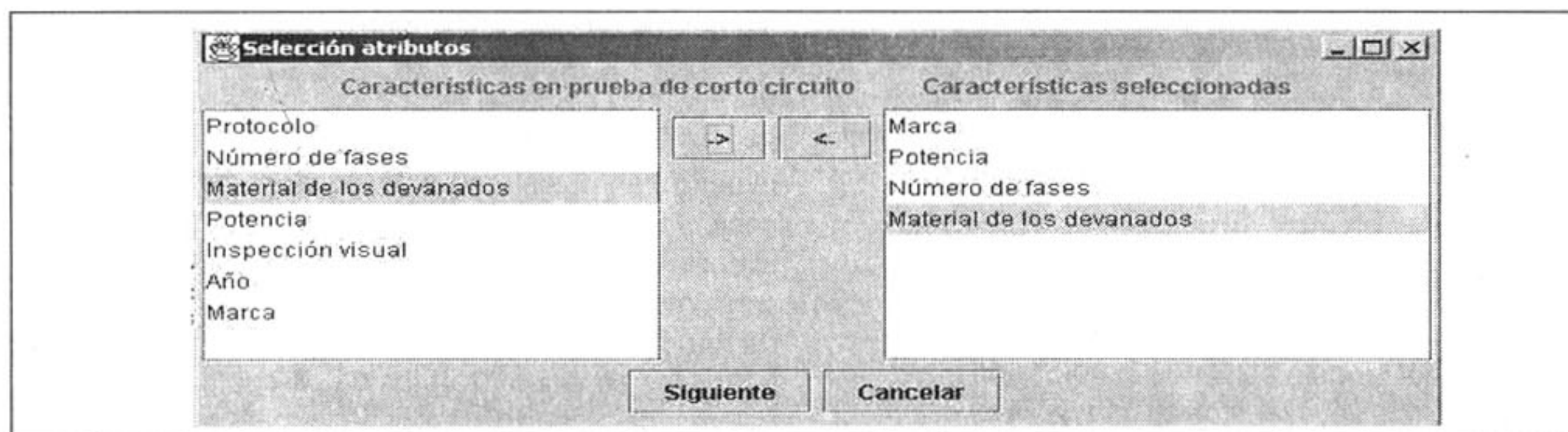


Figura 3. Selección de atributos

Marca	Potencia	Número de fases	Material devana...	Material devana...
A	75	3	COBRE	ALUMINIO
A	25	1	COBRE	COBRE
M	100	3	COBRE	COBRE
C	75	1	COBRE	ALUMINIO

Figura 4. Datos seleccionados de las pruebas de corto circuito.

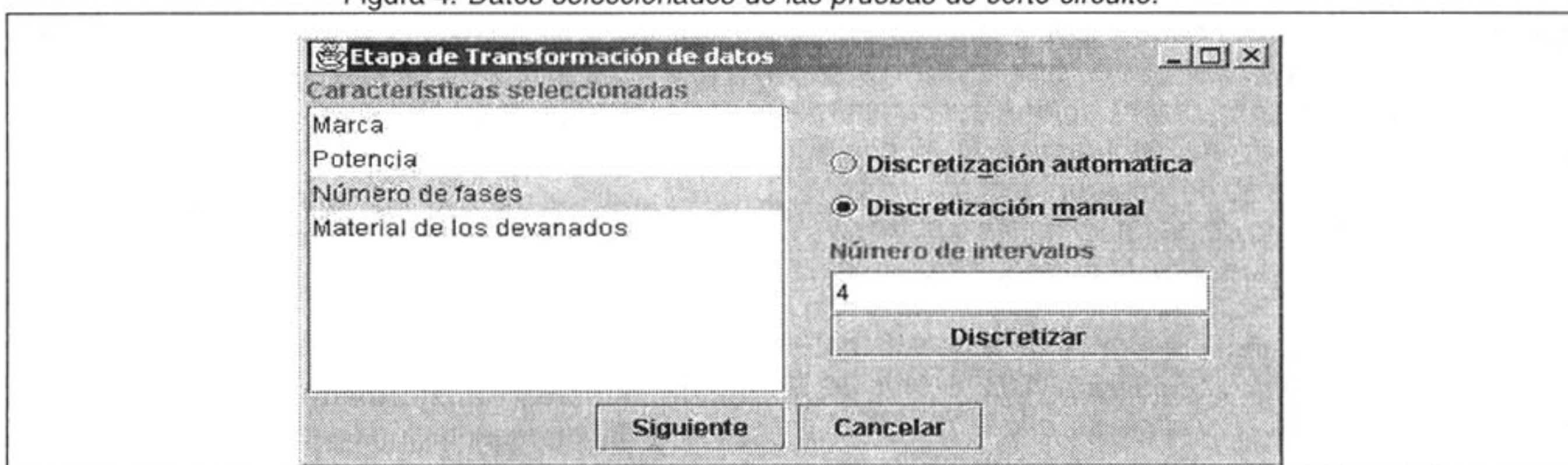


Figura 5. Interfaz para establecer técnica de transformación de datos.

Obtencion de Patrones

Calculo de Reglas de Asociacion

El soporte indica el porcentaje de protocolos que presentan el comportamiento presente en la regla de asociacion.

La confianza es un indicador de la veracidad de la asociacion.

El tamaño indica el número de características presentes en la regla de asociacion.

Soporte Minimo: 0.10

Confianza Minima: 0.80

Tamaño max reglas: 3

Cancelar Siguiente

Figura 6. Interfaz para establecer parámetros para el proceso de minería de datos.

5. RECONOCIMIENTOS

Los autores expresan sus agradecimientos al Instituto Colombiano para el Desarrollo de las Ciencias y la Tecnología "Francisco José de Caldas", COLCIENCIAS (Contrato 292-99) y a la Universidad del Valle por la cofinanciación del proyecto de investigación del cual se deriva este artículo.

6. CONCLUSIONES

- La arquitectura de software propuesta está enfocada a definir elementos que describen el proceso lógico de descubrimiento de conocimiento y es independiente de la forma como se integra al sistema de Gestión de Base de Datos.
- Las arquitecturas de software para descubrimiento de conocimiento permiten identificar componentes del proceso de descubrimiento de conocimiento, lo cual facilita la identificación de patrones reutilizables en otros diseños.
- Cuando las bases de datos se consideran minas de datos, es posible aplicar técnicas de data mining sobre ellas. Esto trae como consecuencia el aumento en el conocimiento del comportamiento del sistema sobre el cual se registran y almacenan los datos.
- Por medio del sistema de descubrimiento de conocimiento en datos de corto circuito, desarrollado para el Laboratorio de Alta Tensión de GRALTA, utilizando la arquitectura propuesta, se ha encontrado conocimiento no evidente en la base de datos descrita por el Modelo de datos

inicial. Este conocimiento permite corroborar hipótesis sobre el comportamiento de los transformadores en las pruebas de corto circuito, formuladas empíricamente.

7. BIBLIOGRAFÍA

- [1.]Agrawal R., Mannila H., Srikant R. Toivonen H., Verkamo A.I., Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining. AAAI Press. The MIT Press, 1996.
- [2.]Agrawal R., Shim K., Developing Tightly-Coupled Data Mining Applications on a Relational Database System. The Second International Conference on Knowledge Discovery and Data mining. Portland Oregon 1996.
- [3.]Bertolino A., Inverardi P., Muccini H., and Rosetti A., "An Approach to Integration Testing Based on Architectural Descriptions," Proc. Third IEEE International Conference on Engineering of Complex Computer Systems (ICECCS97), pp.77-84, Como, Italy, September 1997.
- [4.]Clear J., Dunn D., Harvey B, Heytens M, Lohman P, Mehta A, Melton M, Rohrberg L, Savasere A, Wehrmeister R, Xu M NonStop SQL/MX Primitives for Knowledge Discovery. KDD-99, San Diego, USA 1999.
- [5.]Garlan D., «Software Architecture: a Roadmap,» The Future of Software Engineering, ACM Press, pp.91-101, 2000. J. Zhao, «Applying Slicing Technique to Software Architectures,» Proc. Fourth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS98), pp.87-98, August 1998.
- [6.]Fayyad U., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., . From Data Mining to Knowledge

Discovery and Data Mining. AAAI Press/The MIT Press 1996.

- [7.] Frawley W., Piatetsky-Shapiro G., Matheus C. Knowledge Discovery in Databases: An Overview. In Knowledge Discovery in Databases, Piatetsky-Shapiro G. Frawley W.J. AAAI Press 1991.
- [8.] Han J., Fu, Y., Wang W., Chiang J., Zaiane O., Koperski K. DBMiner: Interactive Mining of Multiple-Level Knowledge in Relational Databases. ACM SIGMOD; Montreal Canada 1996.
- [9.] Han J., Fu Y, Wang W, Koperski K, Zaiane O. DMQL: A data mining query language for relational databases. SIGMOD 96 Workshop.
- [10.] Imielsnki T., Mannila H., Database Perspective on Knowledge Discovery. Communications of the ACM Vol. 39, N° 11, November 1996.
- [11.] Meo R, Psaila G, Cer S. A tightly-coupled architecture for data Mining. ICDE98.
- [12.] Piatestky-Shapiro G. Report on the AAAI-91 Workshop on Knowledge Discovery in Databases, IEEE Expert 6(5), October 1991, pp.74-76.
- [13.] Sarawagi S, Thomas S., Agrawal R., Integrating Association Rule Mining with Relational Database Systems: Alternative and Implication. .Data Mining and Knowledge Discovery, Kluwer Academic Publishers, Vol. 4, 2000.
- [14.] Zhao J., "A Slicing-Based Approach to Extracting Reusable Software Architectures," Proc. 4th European Conference on Software Maintenance and Reengineering, pp.215-223, IEEE Computer Society Press, Zurich, Switzerland, February 2000. ■

AUTORES



María Eugenia Valencia de Abadía. Ingeniera Electricista (1975) y MSc en Ingeniería de Sistemas (1991) de la Universidad del Valle. Profesora titular de la Escuela de Ingeniería de Sistemas y Computación de la Universidad del Valle, Cali, Colombia.
maeva@eisc.univalle.edu.co



Carlos Mauricio Gaona C. Ingeniero de Sistemas de la Universidad Nacional de Colombia. Ingeniero Industrial Universidad Incca de Colombia. Magister en Informática de la Universidad Industrial de Santander (UIS). Ph.D en Computer Science University of Massachusetts. Director del Programa de Ingeniería de Sistemas y profesor asociado de la Escuela de Ingeniería de Sistemas y Computación de la Universidad del Valle, Cali, Colombia.
mgaona@eisc.univalle.edu.co



Marta Elena Millán González. Matemática, Magister en Ingeniería Industrial y de Sistemas de la Universidad del Valle. Ph.D. en Informática de la Universidad Politécnica de Madrid. Profesora Titular de la Escuela de Ingeniería de Sistemas y Computación de la Universidad del Valle, Cali, Colombia.
millan@eisc.univalle.edu.co



Jesús Alexander Aranda Bueno. Ingeniero de Sistemas de la Universidad del Valle. Profesor auxiliar de la Escuela de Ingeniería de Sistemas y Computación de la Universidad del Valle, Cali, Colombia.
jesarana@eisc.univalle.edu.co