# Rasemus: una nueva herramienta de minería de datos*

## [Rasemus: a new data mining tool]

*Ricardo TIMARÁN PEREIRA[1], Ricardo NARVAEZ TERÁN[2]*

***Recibo**: 20.02.2011 - **Ajuste**: 15.03.2011 - **Ajuste**: 20.05.2011 - **Aprobación**: 25.05.2011*

## Resumen

*En este artículo se presenta uno de los resultados del proyecto de investigación cuyo objetivo fue implementar diferentes algoritmos de clustering particional, jerárquico y basado en densidad en una nueva herramienta de minería de datos denominada Rasemus. Esta herramienta permite el Descubrimiento de Conocimiento en Bases de Datos con técnicas de clustering débilmente acoplada con el SGBD PostgreSQL. La arquitectura de Rasemus es modular, compuesta por los módulos de interfaz gráfica, kernel y utilidades, que facilita su mantenimiento y permite la reutilización de sus componentes para incluirlos en otras herramientas de minería de datos. Las pruebas de funcionalidad realizadas en la herramienta Rasemus con los diferentes algoritmos de clustering implementados, demostraron que la herramienta funciona correctamente*

**Palabras Clave:** *Minería de Datos, Algoritmos de Clustering, Herramienta Rasemus*

* Modelo para citación de este artículo monográfico de reporte de caso
TIMARÁN PEREIRA, Ricardo y NARVÁEZ TERÁN, Ricardo (2011). Rasemus: Una nueva herramienta de minería de datos. En: Ventana Informática. No. 24 (ene-jun., 2011). Manizales (Colombia): Universidad de Manizales. p. 25-39. ISSN: 0123-9678

1 Doctor en Ingeniería. MSc. en Ingeniería. Especialista en Multimedia. Ingeniero de Sistemas y Computación.
Director grupo de investigación GRIAS. Profesor Asociado. Departamento de Sistemas. Facultad de Ingeniería. Universidad de Nariño. Correo electrónico: ritimar@udenar.edu.co

2 Ingeniero de Sistemas.
Investigador grupo GRIAS. Departamento de Sistemas. Facultad de Ingeniería. Universidad de Nariño. Correo electrónico: tnricardo@hotmail.com

# Abstract

*This paper presents one of research project results whose objective was the implementation of different partitional, hierarchical and based on density clustering algorithms in a new data mining tool named Rasemus. This is a Knowledge Discovery in Databases tool with clustering techniques, loosely coupled with a PostgreSQL DBMS. Rasemus architecture is modular, composed by Graphic User Interface module, Kernel module and utilities module, which eases maintenance and allows the reuse of components for inclusion in other data mining tools. The functionality tests performed in Rasemus tool with different carried out clustering algorithms showed that the tool works properly.*


**Keywords***: Data Mining, Clustering Algorithms, Rasemus tool.*

# Introducción

El incremento del volumen de información en todo tipo de organizaciones aumenta de una manera considerable cada día. Todo este volumen de información histórica aparte de su función de *"memoria de la organización"* puede ser útil para explicar el pasado, el presente y predecir la información futura utilizando herramientas de descubrimiento de conocimiento (Hernández et al., 2004).

El Descubrimiento de Conocimiento en Bases de Datos (DCBD), es básicamente un proceso automático en el que se combina descubrimiento y análisis (Brachman y Anand, 1994), (Fayyad, Piatetsky-Shapiro y Smyth, 1996a). El proceso consiste en extraer patrones de reglas o funciones, a partir de los datos, para que el usuario los analice (Fayyad, Piatetsky-Shapiro y Smyth, 1996b), (Cabena, Hadjinian, Stadler, Verhees y Zanasi, 1997). Esta tarea implica generalmente preprocesar los datos, aplicar técnicas de minería de datos (*Data Mining*) y presentar resultados (Chen, Han y Yu, 1996), (Imielnski y Mannila, 1996) (Piatetsky-Shapiro, Brachman y Khabaza, 1996). Una de las técnicas más importante de la minería de datos es la agrupación o clasificación no supervisada más conocida como *clustering* (Pérez y Santín, 2006), (Jain, Murty y Flynn, 1999), (Han y Kamber, 2000), (Marcos, 2004). Su objetivo es *particionar* los datos obteniendo el conocimiento de acuerdo a las características de los mismos (Huang, 1997). Los objetos son agrupados basándose en el principio de maximización de similitud dentro de los clústeres y minimización de similitud entre clústeres diferentes (Ng y Han, 1994), (Zhang, Ramakrishnan. y Livny, 1996), para ello utilizan funciones de distancia y similitud o disimilitud dependiendo del tipo de datos a tra-

bajar, ya sean de tipo numéricos o categóricos respectivamente (Han y Kamber, 2000), (Hernández et al., 2004).

Los algoritmos de *clustering* están clasificados en técnicas dependiendo de la forma como se hace la tarea de agrupación, entre ellas se encuentran las técnicas de *clustering* jerárquico, particional, basados en densidad, basados en grillas y otras más (Han y Kamber, 2000), (Hernández, Ramírez y Ferri, 2004). Un método jerárquico crea una descomposición jerárquica de un conjunto de datos, formando un dendograma (árbol) que divide recursivamente el conjunto de datos en conjuntos cada vez más pequeños. Un algoritmo de *clustering* particional obtiene una partición simple de los datos en vez de la obtención de la estructura del clúster tal como se produce con los dendogramas de la técnica jerárquica. Los algoritmos basados en densidad obtienen clústeres basados en regiones densas de objetos en el espacio de datos que están separados por regiones de baja densidad. Los algoritmos de *clustering* basados en grillas cuantifican el espacio en un número finito de celdas y aplican operaciones sobre dicho espacio (Han y Kamber, 2000), (Hernández, Ramírez y Ferri, 2004). Hay que resaltar que *clustering* puede ser una tarea inicial de todo proyecto de minería de datos ya que sirve de soporte a todas las actividades que estos involucran (Bruera, 2001).

Una herramienta DCBD debe integrar una variedad de componentes (técnicas de minería de datos, consultas, métodos de visualización, interfaces, etc.), que juntos puedan eficientemente identificar y extraer patrones interesantes y útiles de los datos almacenados en las bases de datos (Imielnski y Mannila, 1996), (Goebel y Gruenwald, 1999). Las arquitecturas de estas herramientas se pueden ubicar en una de tres tipos: sistemas débilmente acoplados, medianamente acoplados y fuertemente acoplados con un Sistema Gestor de Bases de Datos-SGBD (Timarán, 2001).

En este artículo se presenta uno de los resultados del proyecto de investigación cuyo objetivo fue implementar diferentes algoritmos de *clustering* particional, jerárquico y basado en densidad en una nueva herramienta de minería de datos denominada Rasemus. Esta herramienta permite el descubrimiento de conocimiento en bases de datos con técnicas de *clustering* débilmente acoplada con el SGBD PostgreSQL. Rasemus fue desarrollada, bajo software libre, en el laboratorio KDD del grupo de investigación aplicada en Sistemas GRIAS del Departamento de Sistemas de la Facultad de Ingeniería de la Universidad de Nariño (Colombia). La arquitectura de Rasemus es modular, compuesta por los módulos de interfaz gráfica, *kernel* y utilidades, que facilita su mantenimiento y permite la reutilización de sus componentes para incluirlos en otras herramientas de minería de datos. Las pruebas de funcionalidad

realizadas en la herramienta Rasemus con los diferentes algoritmos de *clustering* implementados, demostraron que la herramienta funciona correctamente.

El resto del artículo está organiza en secciones. En la sección 1, se describe la arquitectura de la herramienta Rasemus y sus diferentes módulos. En la sección 2, se abordan los aspectos de diseño e implementación de la herramienta. En la sección 3, se muestran las pruebas de funcionalidad de los diferentes algoritmos de *clustering* implementados en Rasemus. Finalmente, en la sección 4, se presentan las conclusiones y futuros trabajos.

# 1. Arquitectura de la herramienta Rasemus

La arquitectura de Rasemus es modular, compuesta por los módulos de interfaz gráfica, *kernel* y utilidades que permite soportar las etapas de DCBD: selección, preprocesamiento, transformación, minería de datos y visualización.

Rasemus es una herramienta que brinda al usuario diferentes opciones en técnicas de *clustering*, entre las cuales se pueden encontrar técnicas particionales, jerárquicas y basadas en densidad con los algoritmos básicos de cada técnica. La arquitectura de Rasemus se muestra en la figura 1.

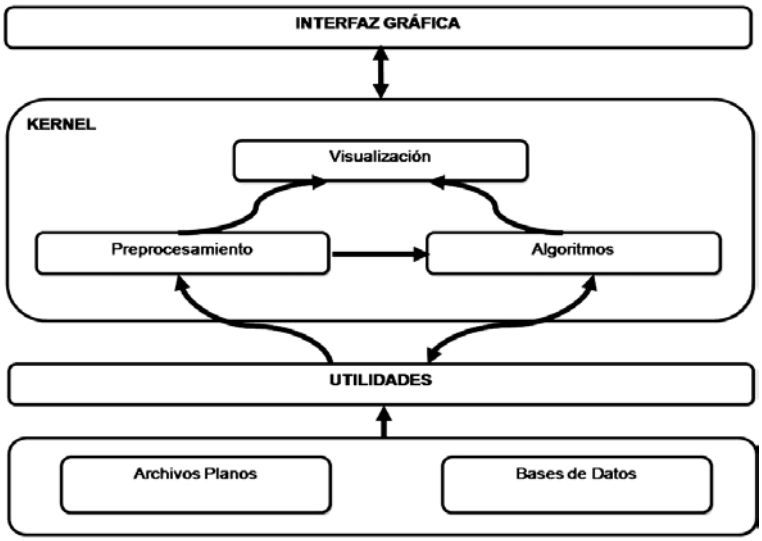


*Figura 1. Arquitectura de Rasemus*

## 1.1 Módulo de utilidades

Este módulo se encarga de establecer la conexión a las diversas fuentes de datos que soporta la herramienta Rasemus: archivos planos, archivos con formato .xls, archivos con formato .arff, y bases de datos. En este módulo se encuentra un conjunto de clases que son utilizadas por las demás clases para desarrollar tareas que son comunes, como dar un formato a los datos o permitir el acondicionamiento a algunos objetos gráficos como tablas y demás recursos visuales. También están las clases que permiten exportar reportes en formatos PDF, RTF y HTML.

## 1.2 Módulo Kernel

En este módulo se encuentran los paquetes principales para la ejecución de tareas de descubrimiento de conocimiento. Se subdivide en los submódulos de preprocesamiento, algoritmos y visualización.

El submódulo de preprocesamiento contiene las diferentes opciones de filtros y rutinas que permiten la adecuación y transformación del conjunto de datos a trabajar, necesarias para lograr mejores resultados con las técnicas de *clustering*. El submódulo de algoritmos contiene las clases encargadas de aplicar las diferentes técnicas de *clustering.* En Rasemus se manejan las técnicas de *clustering* particional con los algoritmos K-means, K-prototype y K-frecuency (Huang, 1997), (Huang, 1998), *clustering* jerárquico del tipo aglomerativo y *clustering* basado en densidad con el algoritmo DBSCAN (Jain, Murty y Flynn, 1999). El submódulo de visualización contiene las clases necesarias para construir y desplegar estructuras que permiten ver de manera gráfica y ágil los resultados de las técnicas de *clustering*.

## 1.3 Interfaz gráfica de usuario

Este módulo da soporte gráfico a todos los demás módulos que lo requieran y se encarga de presentarle al usuario una forma de trabajo amigable. Basado en el concepto de *"arrastrar y soltar"* (*drag and drop*), este módulo facilita el uso de la herramienta para la construcción de experimentos que involucren la interacción entre los diferentes elementos de la ella y el usuario final. En la figura 2 se muestra el entorno gráfico de Rasemus y en la figura 3 los paneles de las diferentes funciones de la herramienta.

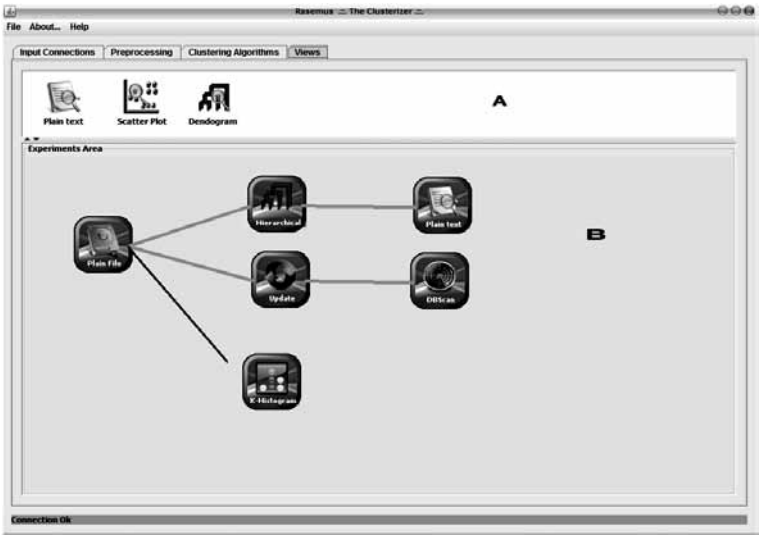


*Figura 2. Entorno gráfico de Rasemus*

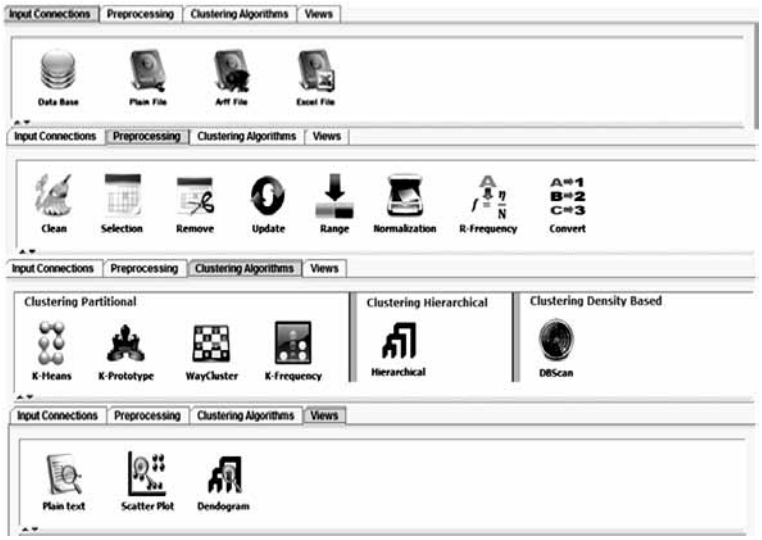


*Figura 3. Funciones de la herramienta Rasemus*

# 2. Aspectos de diseño e implementación de Rasemus

La herramienta Rasemus se diseñó utilizando el análisis y diseño orientado a objetos con el lenguaje de modelamiento unificado UML.

Su diseño es modular con el fin de permitir el crecimiento continuo de esta herramienta. Nuevos filtros, algoritmos y vistas pueden ser fácilmente implementados e incluidos en Rasemus siguiendo este diseño. La estandarización de las entradas de los algoritmos, provenientes del módulo de utilidades o del *kernel*, y sus salidas, hacia los esquemas de visualización, facilitó al grupo de desarrollo la implementación de los algoritmos, trabajando de manera distribuida, haciendo uso de aplicaciones para el control de versiones.

Los requerimientos principales que se tuvieron en cuenta para el desarrollo de Rasemus se muestran en la tabla 1. En la figura 4 se muestra el diagrama de casos de uso principal de Rasemus. En las figuras 5 y 6 se presentan los diagramas de paquetes principal y el de los algoritmos, respectivamente, que ilustran el desarrollo de la herramienta.

**Tabla 1. Requerimientos principales de Rasemus**

| Ref. # | Función | Categoría | Atributo | Detalles y Restricciones | Categoría |
|---|---|---|---|---|---|
| **R1** | Iniciar Herramienta | Evidente | Interfaz | Amigable al usuario | Obligatorio |
| **R2** | Mostrar mensajes de error, respuesta o ayuda cuando sea necesario | Evidente | Interfaz | Cuadros de dialogo o barra de estado | Deseable |
| **R3** | Permitir establecimiento de conexión a fuentes de datos | Evidente | Interfaz | De fácil manejo | Obligatoria |
| **R4** | Permitir la aplicación de filtros | Evidente | Interfaz | | Opcional |
| **R5** | Aplicar técnicas de clustering | Evidente | Interfaz | | Obligatorio |
| **R6** | Permitir visualización de resultados | Evidente | Interfaz | | Obligatorio |
| **R7** | Permitir crear archivos arff con los clúster generados | Oculto | Información | | Opcional |

Rasemus se desarrolló bajo el lenguaje de programación Java™ en su versión 1.6 update 11, lo que la convierte en una herramienta independiente a la plataforma donde se ejecute. Para su construcción, se usaron herramientas de software libre tales como el IDE de desarrollo Netbeans en su versión 6.1 y 6.5, JFreeChart en su versión 1.0.12 con la biblio-

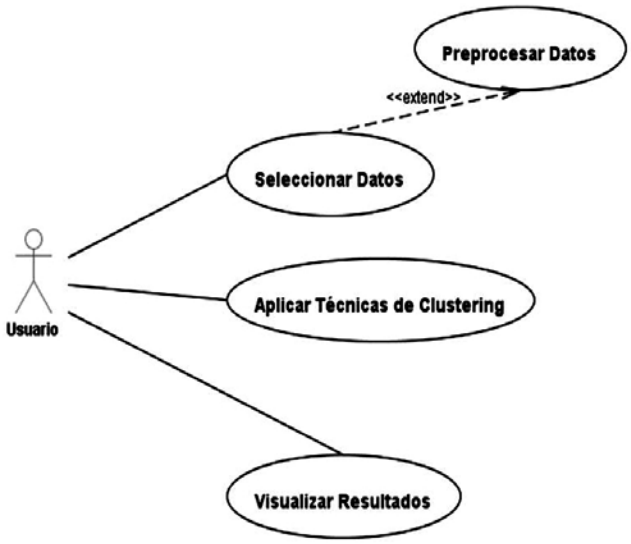


*Figura 4. Diagrama de casos de uso Principal de Rasemus*

teca Jcommon en su versión 1.0.15 para realización de gráficas, API iText en su versión 2.1.5 para la creación de archivos con formatos .pdf y .rtf, JExcelApi 2.6.10 para la lectura de archivos con formato .xls y el sistema gestor de bases de datos PostgreSQL 8.3 para las pruebas de conexión y evaluación.

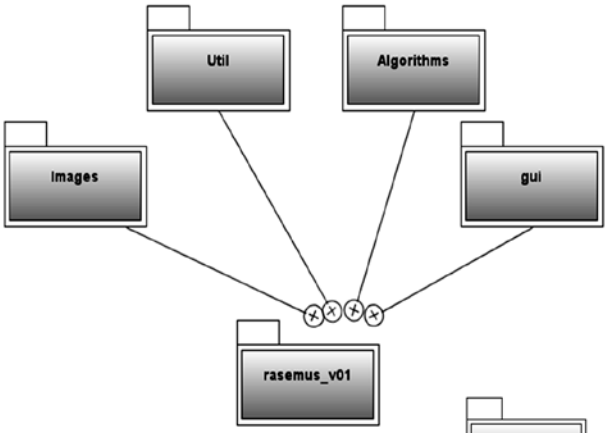


*Figura 5. Diagrama de Paquetes Principal de Rasemus*

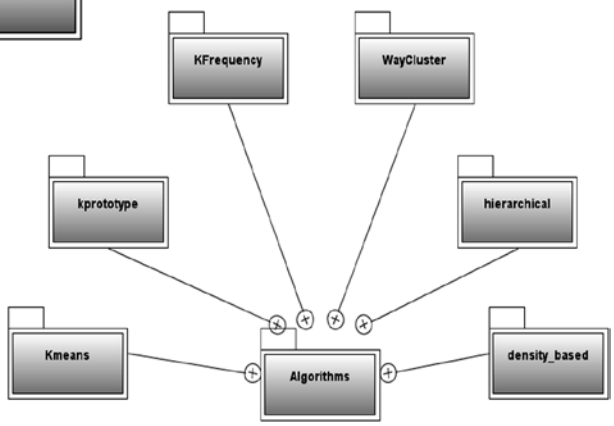


*Figura 6. Diagrama de paquetes de los algoritmos implementados en Rasemus*

# 3. Pruebas de funcionalidad de los algoritmos implementados en la herramienta Rasemus

Para realizar las pruebas de funcionalidad de los algoritmos implementados en la herramienta Rasemus se utilizó un computador Intel® Core™ 2 CPU 4400 @ 2GHZ, Disco Duro de 250 GB, memoria RAM de 3 Gigas y tarjeta de video de 64MB. El objetivo de estas pruebas fue validar que los algoritmos implementados en esta herramienta produzcan iguales resultados (clústeres) que las pruebas manuales realizadas con estos algoritmos. Para realizar las pruebas de funcionalidad se utilizo el conjunto de datos *empleados.arff* compuesto por seis registros y cinco atributos. En la tabla 2 se muestra este conjunto.

**Tabla 2. Conjunto de datos empleados.arff**

| Objeto | Casado | Coche | Hijos | Antiguedad | Sexo |
|---|---|---|---|---|---|
| o0 | Sí | No | 0 | 3 | H |
| o1 | Sí | No | 3 | 2 | M |
| o2 | Si | Si | 2 | 1 | H |
| o3 | No | No | 1 | 1 | M |
| o4 | No | Sí | 1 | 3 | M |
| o5 | Si | Si | 1 | 2 | H |

## 3.1 Pruebas de validación con el algoritmo K-Means

Los parámetros utilizados en esta prueba son: selección de semillas tipo McQueen, número de clústeres a formar 2 (k=2), el método de parada es la convergencia y se selecciona la distancia *Euclidiana* (González y Morales, 2009) para atributos numéricos y para los atributos categóricos la distancia *Hamming.* Contrastando los resultados obtenidos manualmente (tabla 3) con los obtenidos con Rasemus (figura 7), se puede observar que los valores de asignación de clúster son los mismos, lo que demuestra la correcta implementación del algoritmo *k-Means* en Rasemus.

**Tabla 3. Resultado manual de las pruebas con el algoritmo K-means**

| Objeto | Casado | Coche | Hijos | Antigüedad | Sexo | Clúster |
|---|---|---|---|---|---|---|
| o0 | Sí | No | 0 | 3 | H | 0 |
| o1 | Sí | No | 3 | 2 | M | 1 |
| o2 | Si | Si | 2 | 1 | H | 0 |
| o3 | No | No | 1 | 1 | M | 1 |
| o4 | No | Sí | 1 | 3 | M | 0 |
| o5 | Si | Si | 1 | 2 | H | 0 |

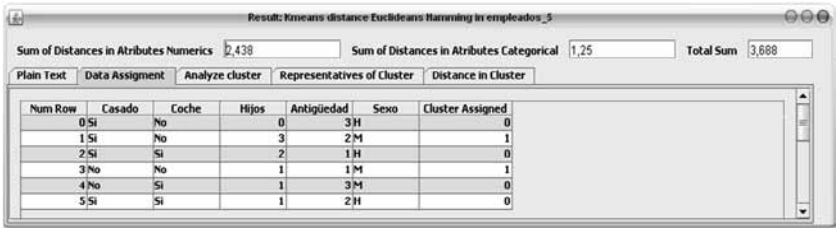


*Figura 7. Resultado de las pruebas con el algoritmo K-means en Rasemus*

### 3.2 Pruebas de validación con el algoritmo KPrototype

Los parámetros utilizados aquí son: número de clústeres a formar 2 (k=2), el método de parada es la convergencia y se selecciona la distancia *Euclidiana* para atributos numéricos y para los atributos categóricos la distancia *Hamming* Comparando los resultados de la prueba manual con el obtenido en Rasemus (figura 8), los valores de asignación de clúster son los mismos, hecho que valida el correcto funcionamiento del algoritmo *KPrototype* en Rasemus

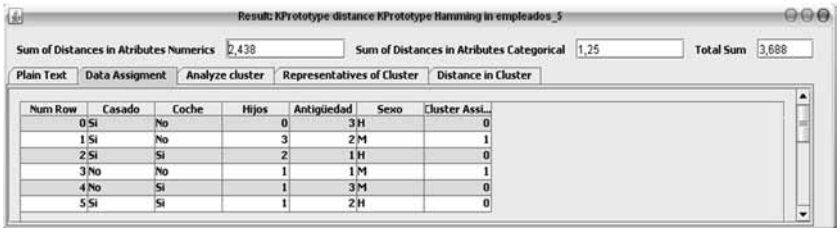


*Figura 8. Resultado de las pruebas con el algoritmo KPrototype en Rasemus*

### 3.3 Pruebas de validación con el algoritmo WayCluster

El algoritmo WayCluster no tiene ningún parámetro para establecer el número de clústeres a formar, lo único que necesita es definir las medidas de distancia y disimilitud. Para este prueba se utilizó la distancia *Euclidiana* y la distancia *Hammig* .De igual manera, como en las pruebas anteriores, los resultados manuales coinciden con los resultados obtenidos en Rasemus (figura 9), lo que demuestra el correcto funcionamiento del algoritmo *WayCluster* en Rasemus.

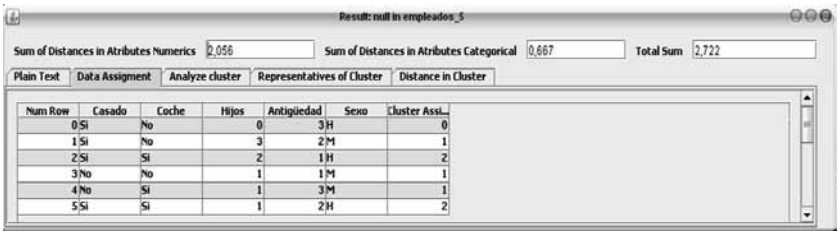


*Figura 9. Resultado de las pruebas con el algoritmo WayCluster en Rasemus*

### 3.4 Prueba de validación con el algoritmo KFrecuency

Los parámetros utilizados aquí son: número de clústeres a formar 2 (k=2), el método de parada es la convergencia y se selecciona solo el tipo de distancia para datos numéricos, la distancia *Euclidiana*. De igual manera, como en las pruebas anteriores, los resultados manuales coinciden con los resultados obtenidos en Rasemus (figura 10), lo que demuestra el correcto funcionamiento del algoritmo *KFrecuency* en Rasemus.

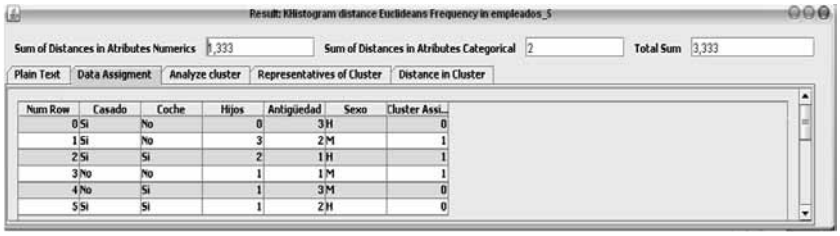


*Figura 10. Resultado de las pruebas con el algoritmo KFrecuency en Rasemus*

### 3.5 Pruebas de validación con algoritmos de *Clustering* jerárquico

Un método jerárquico crea una descomposición jerárquica de un conjunto de datos, formando un dendograma (árbol) que divide recursivamente el conjunto de datos en conjuntos cada vez más pequeños o más grandes. Por la gran funcionalidad de este tipo de algoritmos en aspectos como la clasificación u organización de conjunto de datos, en Rasemus se implementó el algoritmo de *clustering* jerárquico de tipo aglomerativo con diferentes tipos de enlaces. Las Pruebas se realizaron teniendo en cuenta tres tipos de enlace entre clústeres: enlace mínimo (*single link*), enlace máximo (*complete link*) y enlace promedio (*averange link*). Los dendogramas obtenidos en Rasemus que se muestran en las figuras 11,12 y 13 coinciden con los obtenidos manualmente, lo que permite concluir que los algoritmos funcionan correctamente.

## 3.6 Pruebas de validación con algoritmos de Clustering basados en densidad

En las técnicas basadas en densidad, los clústeres se identifican por mirar la densidad entre sus puntos. Regiones con una alta densidad de puntos representan la existencia de clústeres y las regiones con baja densidad de puntos indican el ruido de los clústeres o grupos de valores atípicos. Rasemus implementa los pasos básicos de la agrupación espacial basada en densidad aplicada con el ruido, más conocido como *DBScan*. Este algoritmo es especialmente adecuado para hacer frente a grandes conjuntos de datos, con el ruido, y es capaz de identificar los grupos con diferentes tamaños y formas. Para ello este algoritmo necesita conocer dos parámetros: el primero es el valor conocido como épsilon, que es la medida de densidad que va tener en cuenta para crear un área de barrido y adjuntar los puntos al clúster formado. Otro parámetro necesario es el número mínimo de puntos encontrados en un clúster formado. Es decir, si el número de puntos de un clúster es inferior al mínimo de puntos requeridos el grupo no se tiene en cuenta.

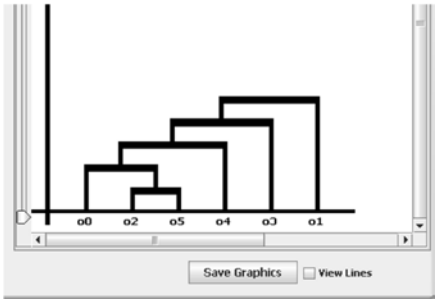


*Figura 11. Resultado algoritmo jerárquico con enlace mínimo en Rasemus.*

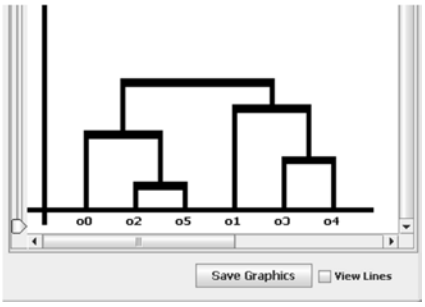


*Figura 12. Resultado algoritmo jerárquico con enlace máximo en Rasemus.*

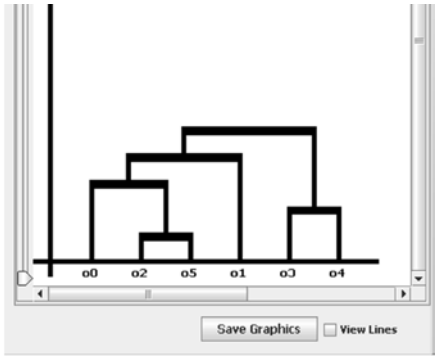


*Figura 13. Resultado algoritmo jerárquico con enlace promedio en Rasemus*

Los parámetros utilizados en las pruebas: son valor mínimo de puntos igual a 1 y épsilon de 3.0, la distancia *Euclidiana* para datos numéricos y la distancia *Hamming* para categóricos y se varió el tipo de barrido. Los resultados de las pruebas en la herramienta Rasemus se muestran en las figuras 14 y 15 que coinciden con las manuales. El clúster -1 indica el ruido.

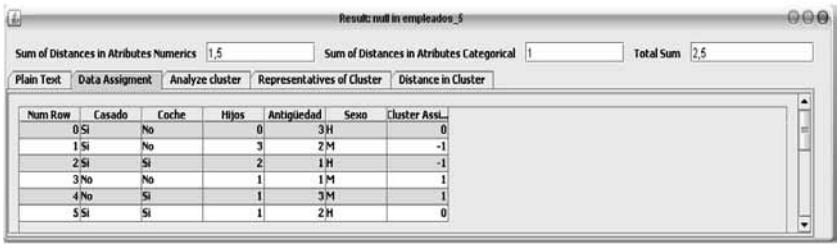


*Figura 14. Resultado algoritmo DBSCAN con barrido directo en Rasemus*

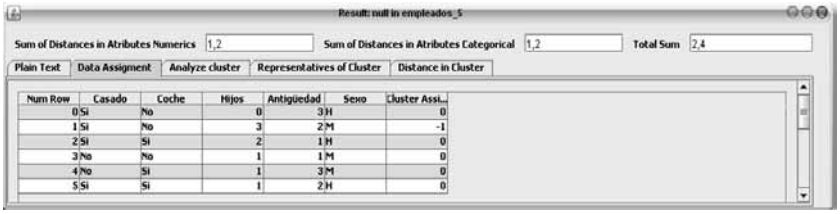


*Figura 15. Resultado algoritmo DBSCAN con barrido conectado en Rasemus.*

# 4. Conclusiones y trabajos futuros

Se cuenta con la primera versión de RASEMUS, una herramienta para el descubrimiento de conocimiento en bases de datos con técnicas de agrupamiento (*clustering*), débilmente acoplada con el SGBD PostgreSQL. Esta herramienta soporta las etapas de minería de datos selección, preprocesamiento, *clustering* y visualización de resultados. En el *kernel* de Rasemus se encuentran implementados tres tipos de algoritmos de *clustering*: particional, jerárquico y de densidad con diferentes tipos de medidas de distancia y similitud, con manejo de datos numéricos, categóricos y mixtos, que permiten al usuario escoger el algoritmo y la métrica que más se adecue a sus necesidades. Las pruebas de funcionalidad realizadas con la herramienta Rasemus demostraron que la herramienta funciona correctamente.

En el proyecto de investigación que dio como resultado RASEMUS se realizaron diferentes pruebas con los algoritmos de *clustering*: particional, jerárquico y de densidad. El análisis de las pruebas realizadas con la herramienta RASEMUS con los diferentes algoritmos de *clustering*: particional, jerárquico y de densidad permite deducir que los algoritmos de tipo particional son eficientes con grandes cantidades de datos, los algoritmos de tipo de datos jerárquico funcionan muy bien con repositorios de datos pequeños, que facilita la construcción de los dendogramas y el *clustering* de densidad sirve para crear segmentos cuando el usuario conoce el grado de similitud de los datos.

Como trabajos futuros están el evaluar Rasemus con otras herramientas del mismo tipo con el fin de determinar su fiabilidad y rendimiento y utilizar la herramienta Rasemus en proyectos reales de minería de datos con empresas colombianas, especialmente con las pequeñas y medianas empresas Pymes.

# Bibliografía

BRACHMAN, R. y ANAND, T. (1994). The Process of Knowledge Discovery in Databases: A First Sketch. En: AAAI-94 Workshop on Knowledge Discovery in Databases, (31/07/1994), Seattle (Washington, USA): AAAI. Proceedings, p. 1-12.

BRUERA, M. R. (2001). Segmentación: Un problema de Minería de datos y dos Algoritmos. En: En Conexión. Año 2. No. 5 (oct-dic.). Buenos Aires (Argentina): IBM Data Management Division Software.

CABENA, P.; HADJINIAN, P.; STADLER, R.; VERHEES, J. and ZANASI, A. (1997). Discovering Data Mining from Concept to Implementation. New York (USA): Prentice Hall. 224 p. ISBN: 978-0137439805.

CHEN, M.; HAN, J. and YU, P. (1996). Data Mining: An Overview from Database Perspectiva. En: Journal IEEE Transactions on Knowledge Data Engineering. Vol. 8, No. 6 (dic.). New Jersey (USA): IEEE Educational Activities Department Piscataway. p. 866-883. ISSN: 1041-4347.

FAYYAD, U.; PIATETSKY-SHAPIRO, G. and SMYTH, P. (1996a). From Data Mining to Knowledge Discovery: An Overview. En: Advances in Knowlege Discovery and Data Mining. Menlo Park, (CA, USA): AAAI Pres/ The MIT Press. 595 p. ISBN: 0-262-56097-6.

FAYYAD, U.; PIATETSKY-SHAPIRO, G. and SMYTH, P. (1996b). The KDD Process for Extracting Useful Knowledge from Volumes of Data. En: Communications of the ACM. Vol. 39, No 11 (nov). New York (NY, USA): ACM. p. 27-34. ISSN: 0001-0782.

GOEBEL, M. and GRUENWALD, L. (1999). A Survey of Data Mining and Knowledge Discovery Software Tools. En: ACM SIGKDD Explorations Newsletter. Vol. 1, No. 1 (jun). New York (NY, USA): ACM. p. 20-33. ISSN: 1931-0145.

GONZÁLEZ, J. y MORALES, E. (2009). Aprendizaje Computacional. [en línea]. San Andrés Cholula, Puebla (México): Instituto Nacional de Astrofísica, Óptica y Electrónica. <http://ccc.inaoep.mx/~emorales/Cursos/NvoAprend/principal.html>.[consulta:10/11/2009].

HAN, J. and KAMBER, M. (2000). Data Mining: Concepts and Techniques. San Diego (CA, USA): Morgan Kaufmann Publishers Inc. 745 p. ISBN: 978-1-55860-901-3.

HERNÁNDEZ, O.J.; RAMIREZ, Q..M. y FERRI, R.C. (2004). Introducción a la Minería de Datos. Madrid (España): Pearson Prentice Hall. 656 p. ISBN: 84-205-4091-9.

HUANG, Z. (1997). A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining. En: Workshop on Research Issues on Data Mining and Knowledge Discovery DMKD 97, (11/05/1997), Tucson (Arizona, USA): ACM SIGMOD. Proceedings.

HUANG, Z. (1998). Extensions to the k-Means algorithm for Clustering Large Data Sets with Categorical Values. En: Data Mining and Knowledge Discovery. Vol 2, No. 3 (sept.). Dordrecht (Holanda): Kluwer Academic Publishers. ISSN 1384-5810.

IMIELNSKI, T. and MANNILA, H. (1996). A Database Perspective on Knowledge Discovery. En: Communications of the ACM. Vol. 39, No.11 (nov): New York (NY, USA): ACM. p. 58-64. ISSN: 0001-0782.

JAIN, A.; MURTY, M and FLYNN, P. (1999). Data Clustering: a review. En: ACM Computing Surveys (CSUR) Surveys. Vol. 31, No. 3 (sept.): New York (NY, USA): ACM. p. 264-323. ISSN: 0360-0300.

MARCOS, M. C. (2004). Browsing y clustering: dos técnicas en auge para la recuperación de información. [en línea]. En ROVIRA, C.; CODINA, L. (dir.) (2004). Documentación digital. Barcelona (España): Sección Científica de Ciencias de la Documentación del Departamento de Ciencias Políticas y Sociales de la Universidad Pompeu Fabra. ISBN: 84-88042-39-6.

NG, R. and HAN, J. (1994). Efficient and Effective Clustering Method for Spatial Data Mining. En: The 20th International Conference on Very Large Data Bases VLDB 94, (12/09/1994), Santiago de Chile (Chile): ACM. Proceedings. p. 144-155. ISBN: 1-55860-153-8.

PÉREZ, C. and SANTÍN, D. (2006). Data Mining Soluciones con Enterprise Miner. México D. F. (México): Alfaomega Ra-Ma. 555 p. ISBN: 970-15-1190-5.

PEÑA, D. (2002). Análisis de Datos Multivariantes. Madrid (España): McGrawHill. 539 p.

PIATETSKY-SHAPIRO, G.; BRACHMAN, R.; KHABAZA, T.; KLOESGEN, W. and SIMOUDIS, E. (1996). An Overview of Issues in Developing Industrial Data Mining and Knowledge Discovery Applications. En: The Second International Conference on Knowledge Discovery and Data Mining KDD 96, Portland (Oregon, USA): ACM. Proceedings AAAI Press. P 89-95. ISBN: 1-57735-004-9.

TIMARÁN, R. (2001). Arquitecturas de Integración del Proceso de Descubrimiento de Conocimiento con Sistemas de Gestión de Bases de Datos: un Estado del Arte. En: Revista Ingeniería y Competitividad. Volumen 3, No. 2 (dic.). Cali (Colombia): Universidad del Valle. ISSN 0123-3033.

ZHANG, T.; RAMAKRISHNAN, R. and LIVNY, M. (1996). BIRCH: An Efficient Data Clustering Method for Very Large Databases. En: ACM SIGMOD International Conference on Management of Data, (4-6/06/1996), Montreal (Canada): ACM. Proceedings. p. 103-114.