Universidad de las Ciencias Informáticas

Facultad 2

Aplicación del algoritmo conceptual RGC en el diseño de Sistemas Basados en Casos

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE

INGENIERO EN CIENCIAS INFORMÁTICAS

Autor:

Osniel Cabrera Frenes

Tutores:

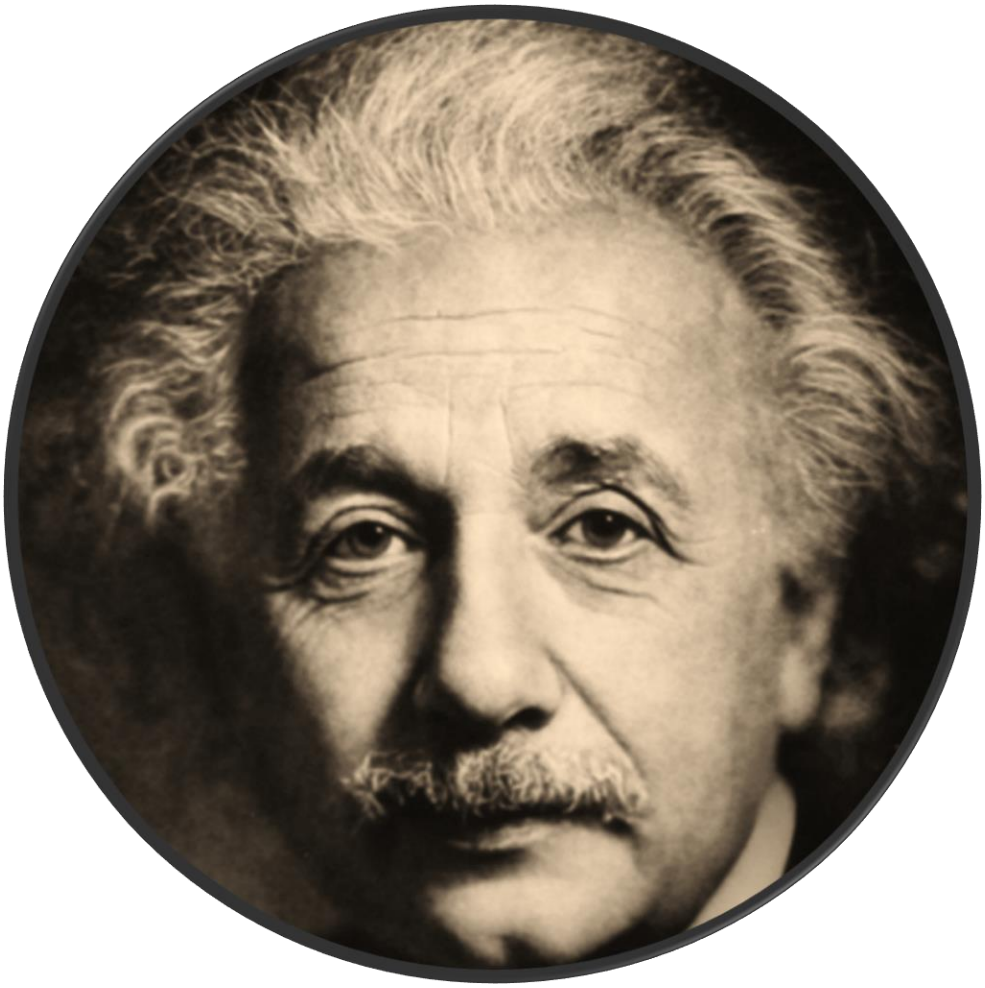
Dra. C. Yunia Reyes González

Dra. C. Natalia Martínez Sánchez

Ms. C. Maidelis Milanés Luque

La Habana, junio 2018

"Año 60 de la Revolución"



"La razón más importante para trabajar en la escuela y en la vida es el placer de trabajar, el placer de su resultado y el conocimiento del valor del resultado para la comunidad."

Albert Einstein

Declaración de autoría

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____

Firma del Autor

Osniel Cabrera Frenes

_____

Firma del Tutor

Dra.C Yunia Reyes González

_____

Firma del Tutor

MSc. Maidelis Milanés Luque

_____

Firma del Tutor

Dra.C Natalia Martínez Sánchez

Datos de contacto

Dra.C. Yunia Reyes González: Graduada de Ingeniera en Ciencias Informáticas en el 2008, actualmente realizando investigaciones en el área de la Inteligencia Artificial, específicamente en Algoritmos de Agrupamiento Conceptual, utilizando el enfoque lógico combinatorio y los Sistemas Basados en el Conocimiento. Profesora de la disciplina de Inteligencia Artificial. Máster en Ciencias y Doctora en Ciencias Técnicas. Vicedecana de Investigaciones y Postgrado de la Facultad 2. Correo electrónico: yrglez@uci.cu

Dra.C. Natalia Martínez Sánchez: Graduada de Licenciatura en Cibernética Matemática en la Universidad Central de Las Villas. Máster en Computación Aplicada y Doctora en Ciencias Técnicas. Profesora Titular. Investiga en el área de la Inteligencia Artificial e Informática Educativa. Ha impartido docencia tanto en pregrado como en postgrado en las ramas de la Inteligencia Artificial, las Matemáticas y Programación. Ha impartido conferencias de pregrado y postgrado en Universidades de Colombia, Perú y Mozambique. Ha participado en eventos nacionales e internacionales, publicando trabajos científicos en revistas y bases de datos de prestigio internacional. Vicerrectora de Formación en la Universidad de las Ciencias Informáticas. Correo electrónico: natalia@uci.cu

Msc. Maidelis Milanés Luque: Graduada de Ingeniera en Ciencias Informáticas en el 2007, actualmente realizando investigaciones en el área de la Inteligencia Artificial, específicamente en el enfoque lógico combinatorio del reconocimiento de patrones y las Redes Neuronales Artificiales. Máster en Ciencias en el año 2017. Profesora de la disciplina de Inteligencia Artificial. Jefa del Departamento de Programación de la Facultad 2. Correo electrónico: mmilanes@uci.cu

Dedicatoria

El presente trabajo de investigación está dedicado a:

Mis padres por su paciencia y compresión.

Mis hermanos por su amor incondicional.

Mis abuelos por su dedicación, amor y cariño.

A la Universidad de las Ciencias Informáticas por haber cumplido mi sueño de ser Informático.

Agradecimientos

Agradezco en primer lugar a mis tres tutoras por su apoyo incondicional sin el cual esta investigación no pudiera haber llegado a su fin.

A mi madre que sin sus consejos y enseñanzas no hubiese alcanzado las metas que hasta ahora me he propuesto.

A mi padre que siempre ha estado ahí en los momentos difíciles.

A mi padrastro por el apoyo que siempre me ha brindado.

A mis abuelos por complacerme en todo lo que mis padres no han querido.

A mis hermanos que a pesar de nuestras peleas siempre nos apoyamos.

A mis tíos que todos de una forma u otra han aportado un granito de arena en mi formación.

A los profesores que me han ayudado en mi formación como profesional en el transcurso de la carrera.

A los compañeros que siempre hemos estado apoyándonos en cada momento en especial a Bienvenido, Wilber, Osciel, Yoany, Herry, Yanlee, Pedro, Alejandro, Jose, Yunior, Raydel, Víctor Alexis, Román, Yaicel, Oswald, Yaidel y Fernando.

A mis compañeras de aula Glenda, Anamelys, Anna Laura, Eileen y Jessica.

A todos,

¡Gracias!

Resumen

Los Sistemas Basados en Casos constituyen un paradigma de la Inteligencia Artificial de gran aplicabilidad en disímiles dominios de la vida real. La organización de la base de conocimiento es una cuestión central en el funcionamiento de este tipo de sistemas, pues de ella depende en gran medida la efectiva recuperación de los casos semejantes al problema a resolver. Es por ello, que en los últimos años la atención en esta temática se dirige hacia este aspecto. La presente investigación propone como objetivo general: aplicar el algoritmo de agrupamiento conceptual RGC en la organización de la base de conocimiento de un Sistema Basado en Casos para contribuir a mejorar la eficacia en la solución de problemas de clasificación. Se describe el algoritmo conceptual RGC en cada una de sus etapas: determinación extensional y determinación intencional; además se propone una estructura jerárquica conceptual para organizar la base de casos que facilita el acceso y recuperación de los casos similares. Para validar los resultados de la propuesta de solución se utiliza el método de validación cruzada y la prueba no paramétrica de Friedman. Se establecen comparaciones en cuanto a la eficacia entre una estructura plana y la estructura jerárquica conceptual propuesta; así como una comparación con otros tipos de organización de la base de conocimiento que utilizan algoritmos como el LC-Conceptual, K-means, Holotipo y el algoritmo Ideal de la Clase. Los resultados demuestran que la solución propuesta garantiza la eficacia en la solución de problemas de clasificación.

Palabras Clave: algoritmos conceptuales, eficacia, Inteligencia Artificial, Sistemas Basados en Casos.

Abstract

The Case Based Systems constitute a paradigm of Artificial Intelligence of great applicability in different domains of real life. The organization of the knowledge base is a central issue in the operation of this type of system, since the effective recovery of cases similar to the problem to be solved depends, to a large extent, on it. That is why, in recent years, attention has been focused on this issue. The general objective of this research is to apply the conceptual grouping algorithm RGC in the organization of the knowledge base of a Case Based System in order to contribute to improve the efficiency in the solution of classification problems. The conceptual algorithm RGC is described in each of its stages: extensional determination and intentional determination; in addition, a conceptual hierarchical structure is proposed to organize the case base that facilitates the access and recovery of similar cases. The cross validation method and Friedman's non-parametric test are used to validate the results of the proposed solution. Comparisons are made as to the effectiveness of a flat structure and the proposed conceptual hierarchical structure; as well as a comparison with other types of knowledge base organization using algorithms such as LC-Conceptual, K-means, Holotype and the Ideal Class algorithm. The results show that the proposed solution guarantees the efficiency in solving sorting problems.

Keywords: Artificial Intelligence, Case Based Systems, conceptual algorithms, efficiency.

Índice

Índice figuras

Índice tablas

Introducción

Los Sistemas Basados en Conocimiento son modelos computacionales de la Inteligencia Artificial que utilizan conocimientos sobre un dominio para arribar a la solución de un problema de ese dominio (Reyes-González 2017).

Una correcta implementación de un Sistema Basado en Conocimiento debe tener en cuenta el proceso de ingeniería del conocimiento, el cual comprende desde la creación de la base de conocimientos hasta el método de solución de problemas. Existen diferentes tipos de representación de un Sistema Basado en Conocimiento como son los Sistemas Basados en Reglas (Ignizio 1991), las Redes Neuronales Artificiales (Kolodner 1992); (Graupe 2013) y los Sistemas Basados en Casos (Kolodner 1992); (Aamodt y Plaza 1994); (Bello 2002) ; (Lopez De Mantaras et al. 2005); (Schank y Riesbeck 2013).

Estos últimos son utilizados para dar solución a problemas como: tareas de diseño (Maher y Pu 2014), aplicaciones de diagnóstico (Singh, Singh y Ahmad 2016), aplicaciones médicas (Blobel 2013), e-learning (Khamparia y Pandey 2017), gestión del conocimiento (Hui et al. 2016), procesamiento de imágenes (Perner 2017), sistemas de recomendación (Sun et al. 2015); (Recio-García, González-Calero y Díaz-Agudo 2014) y sirven de apoyo al proceso de toma de decisiones (Kaklauskas 2015).

Un Sistema Basado en Casos representa el conocimiento en forma de casos y el método de solución de problemas es el Razonamiento Basado en Casos (RBC) (Kolodner 1992); (Aamodt y Plaza 1994); (Schank y Riesbeck 2013). El paradigma del RBC se apoya en casos de problemas resueltos con anterioridad; utilizan una función de distancia o de semejanza para recuperar los casos semejantes al nuevo problema y sus soluciones se reutilizan para obtener una respuesta posible. Posteriormente se realiza el proceso de revisión el cual requiere de un elevado conocimiento sobre el dominio de aplicación y una vez confirmada la solución propuesta, el caso es incorporado a la memoria de ejemplos durante la fase de aprendizaje.

Para lograr un correcto funcionamiento de los Sistema Basado en Casos se hace necesario una eficiente recuperación de los casos semejantes, proceso que está determinado por dos aspectos fundamentales: la estructura de la base de casos y cómo se realiza la comparación entre las descripciones del caso y el problema a resolver. Por lo que se puede afirmar que la recuperación de los casos semejantes está determinada por dos momentos: el primero es el acceso a los casos

y el segundo corresponde a la selección de casos similares.(Richter y Weber 2013); (Fan et al. 2014); (Kang, Krishnaswamy y Zaslavsky 2014); (Li et al. 2015).

La organización de la memoria define el acceso a los casos y de esta depende que la eficiencia en la recuperación no se afecte por el volumen de la experiencia almacenada. El éxito o fracaso de estos sistemas depende en gran medida de una combinación efectiva del acceso a los casos y la selección de los casos similares.

Investigaciones realizadas en este campo (Müller y Bergmann 2014); (Perner 2014); (Sarkheyli y Söffker 2015); (Reyes-González 2017); demuestran que la estructura de la base de casos garantiza una mayor eficiencia en la recuperación de casos similares. La literatura revisada brinda dos enfoques que distinguen el proceso de recuperación de los casos similares: adoptando una estructura plana (Peula et al. 2017) y la organización siguiendo una estructura jerárquica (Herrero et al. 2015); (Guo, Hu y Peng 2014); (Han y Cao 2015); (Fernandes et al. 2016) (Cao et al. 2017). La estructura plana dispone los casos de manera secuencial lo que garantiza encontrar el más similar, pero presenta el inconveniente de la complejidad temporal cuando crece el volumen de datos. Por otra parte, la estructura jerárquica es una alternativa ante el problema de la complejidad temporal, sin embargo, no siempre garantiza encontrar el caso más similar y hace más complejo el aprendizaje incremental de casos solucionados.

Unos de los métodos utilizados por la estructura jerárquica es la utilización de algoritmos de agrupamiento jerárquicos aglomerativos (Aggarwal y Reddy 2013) o divisivos (Goyal y Srivastava 2016), otra variante es la utilización de una estructuración jerárquica conceptual para la base de casos, donde los conceptos constituyen prototipos de subconjuntos de grafos que se determinan calculando el centroide del grupo o seleccionando el medoide como un caso natural representativo del agrupamiento.

Otros antecedentes importantes en el empleo del agrupamiento conceptual lo constituyen las investigaciones de Börner, Wode y Faßauer (1996). Su principal inconveniente es que no puede aplicarse para casos descritos por atributos multi-evaluados y la necesidad de recalcular el grafo cuando se incorporan nuevos casos. Otros autores como: Díaz-Agudo y González-Calero (2001) y Sun et al. (2014); proponen una técnica jerárquica aglomerativa similar a la propuesta por Perner (2006), pero estas dificultan la comprensión de las descripciones conceptuales y consecuentemente afectan el acceso a los casos semejantes, pues la comparación del nuevo problema con los prototipos conceptuales se realiza en función de medidas numéricas y no de las propiedades naturales basadas en los rasgos de los casos.

Por otra parte, las investigaciones de: Huang et al. (2012) y Rezvan, Hamadani y Shalbafzadeh (2013) se refieren a la importancia que merece la selección del conjunto adecuado de rasgos. En esto influye la determinación de sus pesos y las funciones utilizadas para comparar los rasgos y los casos. El Reconocimiento Lógico Combinatorio de Patrones (Ruiz-Shulcloper 2013), ofrece un marco propicio para abordar los aspectos relacionados con la selección de rasgos y permite solucionar problemas relacionados con el agrupamiento de objetos. Por tal motivo, se considera una alternativa a tener en cuenta en la presente investigación.

La mayoría de los métodos de agrupamiento propuestos en la literatura ofrecen estructuraciones de los espacios sobre los que se aplican en forma extensional, es decir, determinan qué objetos están en un cierto agrupamiento. Los algoritmos propuestos por Michalski (1979) se reconocen en la literatura como los primeros en brindar una estructuración conceptual o intencional del espacio, o sea, además de organizar los objetos en grupos, estos pretenden descubrir las características, propiedades o conceptos de estos agrupamientos.

Los algoritmos conceptuales, de manera general, siguen una de las tres aproximaciones siguientes: construyen primero la descripción extensional de los grupos y con base en esta, la descripción intencional (conceptos) de los grupos; determinan primero un conjunto de conceptos presentes en la colección y posteriormente, forman la descripción extensional de cada grupo considerando los objetos que más se ajustan a dichos conceptos o construyen a la misma vez la descripción extensional e intencional de los grupos (Reyes-González 2017).

Basados en los fundamentos del Reconocimiento Lógico Combinatorio de Patrones para problemas no supervisados, se distinguen algoritmos como el LC-Conceptual (Martínez-Trinidad y Ruiz-Shulcloper 1999) y el RGC (Pons-Porrata 2004). Las características distintivas de estos algoritmos constituyen factores a estudiar para la organización jerárquica de la base de casos en un Sistema Basado en el Conocimiento.

El análisis histórico-lógico en torno a los Sistemas Basados en Conocimiento conduce a inferir que aún persisten insuficiencias relacionadas con: la determinación de la función de similitud adecuada al problema a resolver; la selección del conjunto de rasgos y la determinación de su importancia; las estructuras de organización de las bases de casos planas, jerárquicas, jerárquicas conceptuales y el tratamiento inapropiado de rasgos mezclados.

Estos aspectos influyen en que no siempre se proporcione la solución más idónea frente al problema presentado. Todo ello constituye una problemática sobre la cual la ciencia aún no ha

ofrecido conclusiones definitivas por lo que se plantea el siguiente **problema de la investigación:** ¿Cómo contribuir a mejorar la eficacia de los Sistemas Basados en Casos para la solución de problemas de clasificación? Teniendo en cuenta el problema antes propuesto se define como **objeto de estudio:** Sistemas Basados en Casos, enmarcados en el **campo de acción:** estructuras de organización de la base de conocimiento de un Sistema Basado en Casos, utilizando el algoritmo de agrupación conceptual RGC.

Se establece como **objetivo general:** aplicar el algoritmo de agrupamiento conceptual RGC en la organización de la base de conocimiento de un Sistema Basado en Casos para contribuir a mejorar la eficacia en la solución de problemas de clasificación. Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**:

1. Sistematizar los referentes teóricos de la investigación relacionados con los algoritmos conceptuales y los sistemas basados en casos.
2. Aplicar el algoritmo conceptual RGC en el diseño de Sistemas Basados en Casos utilizando una estructura jerárquica conceptual para la base de casos.
3. Valorar los resultados de la aplicación práctica del algoritmo conceptual RGC en el diseño de Sistemas Basados en Casos utilizando una estructura jerárquica conceptual para la base de casos mediante la aplicación de preexperimentos.

Se define como **hipótesis** de la investigación:

Si se aplica el algoritmo conceptual RGC en el diseño de Sistemas Basados en Casos utilizando una estructura jerárquica conceptual para la base de casos se contribuirá a mejorar la eficacia de los Sistemas Basados en Casos para la solución de problemas de clasificación.

Variable independiente: algoritmo conceptual RGC en el diseño de Sistemas Basados en Casos utilizando una estructura jerárquica conceptual para la base de casos.

Variable dependiente: eficacia de los Sistemas Basados en Casos para la solución de problemas de clasificación.

En esta investigación se entiende por eficacia la capacidad del Sistema Basado en Casos para realizar clasificaciones correctas.

Para alcanzar los objetivos propuestos se emplean los siguientes métodos científicos **teóricos y empíricos:**

Métodos Teóricos:

- **Hipotético-Deductivo**: este método es utilizado siguiendo reglas lógicas de deducción para hacer predicciones, que posteriormente son sometidas a verificaciones empíricas.
- **Analítico–Sintético:** este método es utilizado para el análisis de documentos y teorías, permitiendo la extracción de los elementos más importantes que se relacionan con los algoritmos conceptuales para la clasificación no supervisada. Además, es empleado para caracterizar las tecnologías, herramientas y metodologías para la aplicación del algoritmo RGC en Sistemas Basados en Casos.
- **Modelación:** este método es utilizado para representar gráficamente los elementos que componen la propuesta de solución aplicando el algoritmo conceptual RGC y diagramas asociados a la estructura jerárquica conceptual de la base de casos.
- **Sistémico-Estructural-Funcional:** es un método mediante el cual se relacionan hechos aparentemente aislados y se formula una teoría que unifica los diversos elementos.
- **Histórico-Lógico:** este método es utilizado para el estudio del comportamiento del algoritmo conceptual RGC y analizar su funcionamiento con el fin de dar cumplimiento al objetivo de la investigación.

Métodos Empíricos:

- **Medición:** es el procedimiento que se realiza con el objetivo de obtener información numérica acerca de la variable eficacia, donde se comparan magnitudes medibles y conocidas.
- **Observación:** se precisa que parte del fenómeno se va a observar de acuerdo con el objetivo que se persigue con la investigación.
- **Preexperimento:** es utilizado para analizar el comportamiento de la variable eficacia con casos de estudios múltiples.

La investigación se estructura de la siguiente manera: introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y anexos.

El **Capítulo 1** aborda las características de los Sistemas Basados en Casos y las principales limitantes en las estructuras de organización del conocimiento. Además, se estudian los referentes teóricos de los algoritmos conceptuales en el marco del Reconocimiento Lógico Combinatorio de Patrones, como alternativa de integración con los Sistemas Basados en Casos.

En el **Capítulo 2** se describe el algoritmo RGC en el marco del Reconocimiento Lógico Combinatorio de Patrones y cómo este se utiliza para conformar la estructura jerárquica conceptual de la base de conocimientos en un Sistema Basado en Casos.

En el **Capítulo 3** se explican los preexperimentos realizados para la valoración de la propuesta de solución utilizando bases de datos internacionales que permiten comprobar los resultados en cuanto a la variable eficacia.

Las **Conclusiones** resumen los resultados más importantes de la investigación y se explican las **Recomendaciones** para perfeccionarlo en el futuro. Se relacionan, además, las **Referencias Bibliográficas** en las que se sustenta la investigación.

Capítulo 1. Marco teórico referencial sobre los Sistemas Basados en Casos y los algoritmos conceptuales

En el presente capítulo se abordan los principales referentes teóricos de la investigación relacionados con los Sistemas Basados en Casos y los algoritmos conceptuales. Se describen las formas de organización de la base de casos reportadas en la literatura científica, así como su impacto en la recuperación de los casos similares. Se explican los fundamentos del Reconocimiento Lógico Combinatorio de Patrones que sirven de sustento teórico de la investigación. El capítulo finaliza fundamentando la selección del algoritmo conceptual RGC para la solución de la problemática planteada.

1.1. Sistemas Basados en Casos

Los estudios realizados sobre la manera en que los seres humanos toman decisiones constituyen la principal motivación para el desarrollo del enfoque de RBC dentro de la Inteligencia Artificial (Kim, Rudin y Shah 2014). El RBC encuentra sus orígenes en los trabajos de Roger Schank en la Universidad de Yale a finales de la década de los ‘70 e inicios de los ‘80, quien propuso un modelo cognitivo y las primeras aplicaciones de Razonamiento Basado en Casos sustentadas en él (Schank y Abelson 2013). A partir de estas ideas, Janet Kolodner desarrolla el primer sistema con Razonamiento Basado en Casos denominado CYRUS, descrito en (Kolodner 1992) que representa el conocimiento como casos y constituye una implementación del modelo de memoria dinámica propuesto por Schank. Este basamento teórico, sirve de estímulo para varios sistemas que surgen posteriormente, tales como: MEDIATOR, CHEF, PERSUADER y CASEY (Schank y Riesbeck 2013). El interés por el Razonamiento Basado en Casos crece en la comunidad científica mundial, según lo evidencia el establecimiento de una conferencia internacional en la temática a partir de 1995 que se mantiene vigente en la actualidad.

El RBC se inspira en el papel que juega la memoria en la capacidad de razonamiento del hombre. Las definiciones más referenciadas (Kolodner 1993); (Aamodt y Plaza 1994); (Bello 2002); (Lopez De Mantaras et al. 2005); (Schank y Riesbeck 2013) en la literatura clásica del tema coinciden en afirmar que el RBC es un paradigma de la Inteligencia Artificial el cual representa el conocimiento como casos, permitiendo abordar problemas nuevos mediante la reutilización de las soluciones dadas a situaciones similares ocurridas con anterioridad. Es particularmente adecuado en dominios para los cuales las experiencias humanas desempeñan un rol significativo en la resolución de problemas (Shokouhi, Skalle y Aamodt 2014).

Para la presente investigación se consideran significativos los argumentos expuestos en Pal, Dillon y Yeung (2012) y Richter y Weber (2013), en cuanto a las posibilidades que puede ofrecer el RBC, sobre otros enfoques de la Inteligencia Artificial, cuando este se utiliza de manera apropiada.

El uso del RBC simplifica la tarea de adquisición del conocimiento, pues sólo se necesita la recopilación, representación y almacenamiento de experiencias relevantes ya existentes, evitando así la formalización del conocimiento usando reglas, frames o transformando ese conocimiento en probabilidades o pesos de una red neuronal artificial. Utiliza directamente el conocimiento almacenado en casos o ejemplos resueltos con anterioridad. Evita repetir errores cometidos anteriormente, pues el sistema puede usar la información sobre lo que causó fallas en el pasado para predecir fallas en el futuro. Es aplicable en dominios que no son totalmente comprensibles, definidos y modelados e incluso con datos incompletos e imprecisos. Posibilita realizar predicciones en cuanto al probable éxito de una solución y posee la capacidad de aprendizaje en todo momento, propiciando nuevas soluciones frente a situaciones nuevas. Provee un medio para la explicación de las soluciones y permite razonar en dominios con pequeño cuerpo de conocimientos (Pal, Dillon y Yeung 2012).

La metodología del RBC proporciona un modelo computacional cercano al razonamiento de las personas, que es bastante intuitivo y fácil de comprender. En consecuencia, cuando se implementa, utiliza un paradigma humano en un contexto computacional; beneficiándose de la vasta memoria y la velocidad proporcionada por las computadoras. Es capaz de tratar cuestiones informales, por lo que no requiere una formalización extensa y compleja de los problemas (Richter y Weber 2013).

La amplia aplicabilidad del RBC en diversas áreas del conocimiento y para disímiles propósitos, entre los que se destacan tareas de análisis: clasificación, diagnóstico y predicción y tareas de síntesis relacionadas con la planificación, configuración y diseño (Maher y Pu 2014), permiten sustentar la idea de que el RBC es un valioso ejemplo de sistemas de apoyo a las decisiones (Sharaf-El-Deen, Moawad y Khalifa 2014).

1.2. Estructuras de organización de la base de casos

La base de casos es uno de los componentes más importantes para el proceso del RBC (Perner 2014); (Sarkheyli y Söffker 2015). El modo en que se estructura la memoria define cómo se accede a los casos e influye en la recuperación, de ahí la importancia que, en los últimos años, se concede a estos aspectos.

En Richter y Weber (2013) se reconocen tres formas principales de organizar la base de casos: plana, estructurada y no estructurada (para la representación de textos). La organización plana es la más simple de diseñar e implementar, y la más adecuada para un pequeño número de casos. Estos se disponen de forma secuencial en una lista, arreglo o archivo y para realizar la búsqueda se accede a cada uno de ellos, determinándose el caso o el conjunto de casos que mejor igualan la entrada. Sin embargo, cuando aumenta la cantidad de datos a procesar se presenta el inconveniente de la complejidad temporal que produce ineficiencia en la recuperación (Peula et al. 2017). Las propuestas de Chattopadhyay et al. (2013); Rezvan, Zeinal Hamadani y Shalbafzadeh (2013); Guessoum, Laskri y Lieber (2014); Kocsis et al. (2014); Fan et al. (2014); Ince (2014); Hu, Qi y Peng (2015); Liang et al. (2015); Platon, Dehkordi y Martel (2015) y García, Trujillo y Arza (2016) asumen una estructura plana.

Como alternativa se necesita de una representación estructurada para los casos, la cual puede comprender organización en redes y jerarquías.

Si se utilizan redes de discriminación (Aamodt y Plaza 1994) cada nodo es una pregunta que subdivide al conjunto de casos, cada subnodo es una respuesta distinta a la pregunta y las preguntas más importantes se colocan en la parte superior de la jerarquía. Este enfoque requiere mucha memoria para almacenar la red y los procedimientos para agregar nuevos casos son muy costosos; ya que la jerarquía necesita ser reestructurada cada vez que se incorpora un nuevo caso.

El empleo de árboles de decisión, en los cuales las hojas contienen todos los casos y los nodos intermedios contienen particiones de la base original es otra de las propuestas para organizar las bases de casos (Banerjee y Chowdhury 2015). Este tipo de enfoque es particularmente útil cuando las bases de casos son grandes, pero cuando los casos no están completamente disponibles y el dominio no está bien definido resulta más difícil de aplicar.

Un método alternativo, son los modelos basados en ejemplares (Aamodt y Plaza 1994); (Branting 2014) que no necesariamente requieren todos los rasgos o todos los casos por adelantado. Sin embargo, buscar un criterio para determinar qué es un buen ejemplar no resulta una tarea trivial.

La estructura jerárquica, es de las más utilizadas (Guo, Hu y Peng 2014). En ella los casos se localizan en nodos de un árbol o grafo acíclico dirigido, el grafo subdivide los casos de acuerdo a los atributos que comparte y localiza atributos comunes en los nodos internos del árbol. De esta forma todos los casos que comparten valores se localizan bajo dicho nodo. Esta jerarquía permite

una búsqueda más eficiente ya que se sigue por un determinado camino en dependencia de los valores de los rasgos predictores del nuevo problema.

Una tendencia actual en esta dirección consiste en la aplicación de técnicas de agrupamiento para fraccionar la base de casos en partes más manejables. En Martínez Sánchez (2009) la base de casos se define mediante una estructura jerárquica de grupos, representados por el elemento de mayor tipicidad de los mismos denominado Holotipo (Martínez y Guzmán 2001) del conjunto, en un nivel superior. Sin embargo, este holotipo al ser una instancia en particular del agrupamiento no representa en su totalidad las características de todos los casos de su grupo.

El algoritmo k-means y sus variantes difusas, es ampliamente utilizado en esta dirección (Han y Cao 2015); (Fernandes et al. 2016); (Cao et al. 2017). Por lo general, k-means, utiliza una función de distancia para calcular el centroide del grupo, considerando únicamente valores numéricos, por lo que, ante la problemática de variables no numéricas, estas son codificadas con la consecuente pérdida de información que implica. El empleo del algoritmo k-means presupone que el espacio de representación de los objetos es métrico, además de que requiere de semillas para su funcionamiento y por otra parte resulta inadecuado para grupos no convexos, de tamaños y densidad diferentes.

En tanto Mittal, Sharma y Dalal (2014); Müller y Bergmann (2014); Zhong, Xie y Lin (2015); Zhu et al. (2015); Aliaga et al. (2015), utilizan algoritmos de agrupamientos tanto jerárquicos como particionales, algunos incluyen variantes difusas, para la organización de la base de casos. Por lo general, calculan o seleccionan un caso como prototipo representativo de cada agrupación, que en ocasiones es un caso real dentro del grupo, pero en otras es el centroide o medoide. Estos prototipos están basados en medidas numéricas como la media o mediana, lo que implica que se consideren sólo rasgos numéricos en los casos o la transformación de valores categóricos a numéricos. En el caso de los algoritmos jerárquicos presentan dificultades para realizar el aprendizaje incremental pues necesitan reajustar la estructura de organización cuando se incorpora un nuevo caso a la jerarquía.

Como alternativa, se propone en Perner (2006); Jänichen y Perner (2006) y Bichindaritz (2008) una estructuración jerárquica de tipo conceptual para la base de casos. Los conceptos constituyen prototipos de subconjuntos de grafos que representan una abstracción de cada grupo de casos similares. Las propuestas referenciadas se encuentran en el dominio de aplicación de la visión por computador y la interpretación de imágenes médicas; consideran, básicamente dos opciones para determinar los prototipos conceptuales. Uno de ellos calcula un caso artificial situado en el

centroide del grupo y el otro consiste en seleccionar el medoide como un caso natural representativo del agrupamiento.

Los trabajos de Börner, Wode y Faßauer (1996); Díaz-Agudo y González-Calero (2001) y Sun et al. (2014) constituyen importantes antecedentes en el empleo del agrupamiento conceptual para la organización de la base de casos. En la investigación de Börner, Wode y Faßauer (1996) se construye un grafo de casos utilizando el algoritmo Galois y el análisis de conceptos formales. Sus limitantes están relacionadas con la necesidad de reestructurar el grafo cuando se incorporan nuevos casos y su imposibilidad para manipular atributos multi-evaluados. Por su parte en Díaz-Agudo y González-Calero (2001) se generan conceptos basados en probabilidades aplicando una técnica jerárquica aglomerativa similar a la propuesta en Perner (2006). En Sun et al. (2014) se organiza la base de casos mediante la aplicación del algoritmo jerárquico incremental COBWEB (Fisher 1987) que formula los conceptos en función de las probabilidades de los atributos. Este algoritmo maneja sólo valores cualitativos, por lo que para trabajar con valores numéricos estos deben ser discretizados.

Las variantes conceptuales presentadas, si bien superan las deficiencias del agrupamiento convencional y permite reducir el esfuerzo requerido en la recuperación de los casos, aún presenta importantes limitantes. Por lo general, los conceptos considerados como prototipos son casos artificiales que tienen asociados el cálculo de varianzas, medias, distancias o probabilidades. Estas medidas estadísticas dificultan la comprensión de las descripciones conceptuales obtenidas y consecuentemente afectan el acceso a los casos semejantes, pues la comparación del problema a resolver con los prototipos conceptuales se realiza en función de medidas numéricas y no de las propiedades naturales basadas en los rasgos de los casos. En las estrategias incrementales, por su parte, debe señalarse el hecho de que la estructuración resultante depende del orden de presentación de los casos y del umbral seleccionado para generar los agrupamientos, lo cual repercute en la organización de la memoria de ejemplos. No obstante, en Perner (2014) se asegura que la descripción de los conceptos, los operadores para su construcción, así como la función de evaluación de conceptos están en el centro de la investigación en la estructura de memoria conceptual.

En las formas organizativas de las bases de casos no estructuradas, se destacan las ontologías como aplicación en los dominios correspondientes a sistemas de recuperación de información, donde prevalece la documentación textual poco estructurada. En Amailef y Lu (2013); Delir Haghighi et al. (2013); Zidi et al. (2014), El-Sappagh y Elmogy (2015); Maalel, Mejri y Ghézala

(2016) y Chen et al. (2016) se describen sistemas basados en casos que utilizan ontologías para representar el conocimiento.

Por lo general, el propio modelo de organización de la base de casos determina el acceso y recuperación del conjunto de casos candidatos. En una estructura plana ambos procesos se afectan por la disposición secuencial de los casos en tanto en una estructura jerárquica se facilita la búsqueda (ver figura 1).

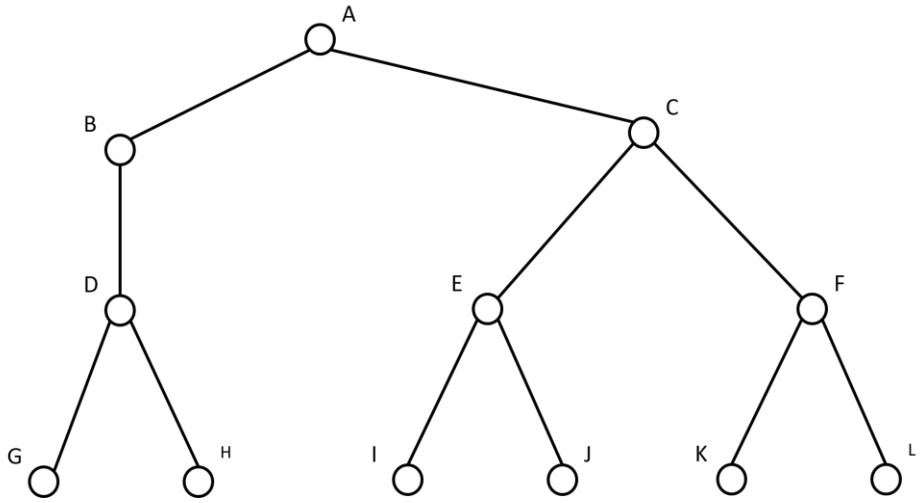


Figura 1 Organización jerárquica de la base de casos que favorece el acceso y recuperación. Fuente (Reyes-González 2017)

1.3. Reconocimiento Lógico Combinatorio de Patrones

Para representar los objetos en el Reconocimiento de Patrones, se consideran básicamente dos variantes según lo planteado en Ruiz-Shulcloper (2009). Una de ellas en términos de un cierto alfabeto de partes (primitivas) de los objetos, característico del enfoque sintáctico estructural. La otra, en términos de un conjunto de rasgos (propiedades, características o variables), típico del enfoque estadístico y de otras aproximaciones como el lógico combinatorio. En esta última forma de representación, por exigencias de los modelos matemáticos empleados, los objetos son representados como una secuencia de números o exclusivamente de valores no numéricos. Sin embargo, en muchos problemas reales que con frecuencia se presentan en las denominadas disciplinas poco formalizadas del conocimiento, la Medicina, Sociología, Geociencias, Criminalística, entre otras, coexisten ambos tipos de descripciones e incluso con valores ausentes. Se les conoce como descripciones mezcladas, pues aparecen simultáneamente rasgos numéricos y no numéricos, e incompletas porque algunos valores son desconocidos.

El Reconocimiento Lógico Combinatorio de Patrones encuentra su basamento teórico matemático en la Lógica Matemática, la Teoría de Testores, la Teoría Clásica de Conjuntos, la Teoría de los Subconjuntos Difusos, la Teoría Combinatoria, y la Matemática Discreta en general. Las ideas centrales de este enfoque consisten en suponer que los objetos se describen por medio de una combinación de rasgos numéricos y no numéricos, y los distintos valores pueden ser procesados por funciones numéricas (Ruiz-Shulcloper, Guzmán Arenas y Martínez-Trinidad 1999).

Este enfoque presupone que el espacio en que se modelan los objetos son en general productos cartesianos de los conjuntos de valores admisibles de las variables en términos de las cuales se describen todos los objetos, no espacios métricos. Además, tiene en cuenta el concepto de analogía o similitud que se refiere al parecido que poseen entre sí dos objetos en dependencia de los rasgos que los describen. En este sentido se considera también la semejanza o parecido que guardan dos valores de un mismo rasgo, lo que se conoce como criterios de comparación de valores de las variables o función de comparación por rasgos (Ruiz-Shulcloper, Guzmán Arenas y Martínez-Trinidad 1999).

En los problemas de selección de rasgos con datos mezclados e incompletos, la principal herramienta empleada es la Teoría de Testores. Se denomina testor de una matriz de entrenamiento (ME), al subconjunto de rasgos $T \subseteq \Re$, tales que al eliminar todas las columnas de ME excepto las de T, no aparecen nuevas sub-descripciones semejantes en clases diferentes (Alba-Cabrera, Ibarra-Fiallo y Godoy-Calderon 2013); (Sanchez-Diaz et al. 2014).

Un testor se denomina irreducible (típico) si al eliminar alguna de sus columnas deja de ser testor (Alba-Cabrera, Ibarra-Fiallo y Godoy-Calderon 2013); (Sanchez-Diaz et al. 2014). En el cálculo de los testores típicos se utilizan algoritmos tales como el LEX, CT_Ext, BR, fast-CT_Ext y Fast-BR (Lias-Rodriguez y Sanchez-Diaz 2013).

El marcado carácter diferenciante e irreducible de los testores típicos, y por ende de los rasgos que lo conforman condujo a formular la definición de peso informacional de un rasgo (Ruiz-Shulcloper 2009), en función de la frecuencia relativa de aparición de ese rasgo en la familia de los testores típicos y la longitud de los testores típicos en los que aparece el rasgo, según la ecuación 1:

$$\varepsilon(x_i) = \alpha F(x_i) + \gamma L(x_i) \qquad (1)$$

Los parámetros $\alpha > 0$, >O y $\alpha+\gamma=1$, α y γ ponderan la influencia de Fi (frecuencia de aparición) y Li (longitud de los testores) respectivamente y se determinan por el experto del área de aplicación.

Un problema de clasificación supervisada dentro del Reconocimiento Lógico Combinatorio de Patrones consiste en, construir un algoritmo que permita, a partir de una muestra no vacía de objetos estructurados en clases, decidir a cuál de las clases pertenece un objeto nuevo que se quiera clasificar. Los algoritmos basados en el ideal de la clase (AIC), en umbrales de exactitud (AUE), tipo votación (ATV), basados en la tipicidad y el contraste (ATV), en conjuntos de representantes (CR) y el modelo de algoritmos Kora-Ω son ejemplos de métodos de clasificación supervisada (Ruiz-Shulcloper 2009).

Resolver un problema de clasificación no supervisada consiste en determinar la estructura interna de un conjunto de descripciones de objetos en el espacio de representación. Esta estructura interna depende en una primera instancia, de la selección del propio espacio de representación y de la forma en que los objetos se comparen, es decir, del concepto de similitud que se emplee y del criterio de agrupamiento que se utilice. En este sentido se pueden encontrar dos situaciones diferentes: una en la que por determinadas razones se conoce que los objetos se agrupan en un número dado de clases, pero no se tiene muestra alguna de este (agrupamiento o estructuración restringida) y otra en la que no se cuenta con esa información (agrupamiento o estructuración libre).

Los algoritmos de agrupamiento, convencionales o tradicionales como también son conocidos en la literatura, presentan importantes limitaciones identificadas en Michalski y Stepp (1981): dejan el problema de la interpretación de los grupos al analista de datos. No tienen en cuenta los métodos que las personas emplean para agrupar objetos. La lógica humana tiende a agrupar objetos en categorías caracterizadas por conceptos, en grupos similares teniendo en cuenta algún atributo relevante o más importante que el resto. Los métodos tradicionales de agrupamiento no toman en consideración ningún concepto o construcciones lingüísticas que las personas usan para describir colecciones de objetos. Para enfrentar estas limitaciones a finales de los años 70 e inicios de los 80, Rysazard S. Michalski introdujo un conjunto de ideas que dieron origen al agrupamiento conceptual (Michalski 1980).

1.3.1. Selección de rasgos

La selección de rasgos es uno de los pasos fundamentales en cualquier problema de clasificación debido a que la mayoría de los problemas de reconocimiento de patrones están basados en la descripción de los objetos en términos de un conjunto de rasgos. En la literatura se identifican dos problemas diferentes pero muy vinculados: la selección de rasgos para la clasificación y la selección de rasgos para la descripción (Ruiz-Shulcloper 2009).

La selección de rasgos para la clasificación, es la determinación del mejor subconjunto de rasgos para la clasificación de nuevos objetos (no clasificados). Esto conlleva la reducción del conjunto de todos los posibles rasgos (reducción de la dimensionalidad) sobre la base de las diferencias que estos rasgos presentan para clasificar a los nuevos objetos y otros problemas de optimización adicionales del subconjunto de rasgos a emplear (Ruiz-Shulcloper, 2009).

1.3.2. Clasificación supervisada

Un problema de clasificación supervisada consiste en, dado un universo de objetos estructurados en clases, de cada una de las cuales se tiene una muestra no vacía, que permite construir un algoritmo a partir de esta muestra, decidirá a cuál de las clases pertenece un objeto nuevo que se quiera clasificar.

Un problema de Reconocimiento de Patrones en la clasificación supervisada es un procedimiento efectivo donde la clasificación de un objeto O∈U por un clasificador supervisado A, se entiende la acción de asignar un r-uplo de pertenencias a las clases de U, tomando dicho objeto como entrada de A. En algunos textos también se entiende por clasificación el resultado de esta acción. La clase que A, con matriz de entrenamiento M, le asigna a un objeto O se denotará por αA (M, O). A esto último se le llamará r-uplo de pertenencia del objeto asignado por A. En la descripción de los algoritmos se utiliza la notación funcional de un clasificador supervisado (Ruiz-Shulcloper 2009).

1.3.3. Clasificación semisupervisada

Los problemas de clasificación semisupervisada combinan tanto la clasificación supervisada como la no supervisada, su definición formal plantea: la clasificación semisupervisada se considera una extensión de la clasificación supervisada donde el conjunto de entrenamiento está formado por un conjunto L de objetos clasificados y un con un conjunto U de objetos sin clasificar, donde se asume que el número de objetos no clasificados es mayor que los clasificados.

El objetivo de la clasificación semisupervisada es entrenar un clasificador f a partir de los conjuntos L y U, de manera que se podrá obtener una clasificación más exacta que la cantidad de objetos ya clasificados por los problemas de clasificación supervisados.

1.3.4. Clasificación no supervisada

En la clasificación no supervisada o agrupamiento el propósito es juntar (agrupar) los objetos según su analogía (parecido, semejanza, cercanía si se está hablando de un espacio de representación con distancia definida). Los tres elementos esenciales que lo constituyen son: el

espacio de representación de los objetos, la medida de similitud (β, función de semejanza) y el criterio de agrupamiento Π, es decir, la manera en que es utilizada la similitud para la solución del problema planteado (Reyes-González 2014).

El objetivo de los algoritmos de agrupamiento es dado un conjunto de objetos definidos en términos de un conjunto de rasgos, intentar construir particiones o cubrimientos de este conjunto, donde la semejanza intragrupo sea máxima y la semejanza inter-grupos sea mínima (Reyes-González et al. 2016).

Según Reyes-González (2014) los algoritmos de agrupamiento convencionales tienen las siguientes limitaciones:

1 Dejan el problema de la interpretación de los grupos al analista de datos.
2 No tienen en cuenta los métodos que los humanos emplean para agrupar objetos. Las personas tienden a agrupar objetos en categorías caracterizadas por conceptos, en grupos similares teniendo en cuenta algún atributo relevante o más importante que el resto.
3 No toman en consideración ningún concepto o construcciones lingüísticas que las personas usan para describir colecciones de objetos (Michalski y Stepp 1981).

1.4. Algoritmos conceptuales

Los algoritmos de agrupamiento conceptual se componen de dos tareas fundamentales, las cuales no tienen necesariamente que ser independientes ni realizarse en un orden determinado (Reyes-González 2017):

1 La estructuración o determinación extensional: se lleva a cabo el proceso de agrupar entidades, en el que se determinan grupos a partir de una colección de objetos, esto no es más que la enumeración de los objetos que lo componen los grupos.
2 La caracterización o determinación intencional: se determina el concepto de cada grupo de la estructuración, las propiedades que caracterizan el agrupamiento.

Los algoritmos de agrupamiento conceptual se pueden dividir en dos grandes grupos, los algoritmos incrementales y los no incrementales. Los algoritmos incrementales basan su funcionamiento en la adaptación de los agrupamientos (o conceptos) con los nuevos objetos que se le van presentando, es decir, cada vez que llega un nuevo objeto mediante una cierta estrategia éste es clasificado en los agrupamientos ya existentes o se crean nuevos agrupamientos. Por otro

lado, los algoritmos no incrementales estructuran una muestra de objetos sin presuponer que éstos llegan de uno en uno.

En los trabajos de Ruiz-Shulcloper (2009) y Reyes-González (2017) se realiza un análisis crítico de diferentes algoritmos de agrupamiento conceptual, atendiendo a sus características y funcionamiento. Destacándose entre sus principales resultados los siguientes:

1 La principal desventaja de los algoritmos conceptuales de tipo incremental es la dependencia del resultado (la estructuración) en función del orden de presentación de los objetos al algoritmo. Mientras que en los de tipo no incremental se determina el número de las agrupaciones de manera aleatoria, lo que constituye una dificultad en la vida real, debido al desconocimiento de los posibles agrupamientos en ciertos tipos de problemas.
2 Los algoritmos que no pertenecen al enfoque lógico combinatorio del reconocimiento de patrones construyen sus conceptos en función de criterios probabilísticos o estadísticos, por lo que su interpretación puede ser engorrosas para personas no especializadas en esas áreas del conocimiento.
3 Los algoritmos LC-Conceptual y RGC (ambos pertenecientes al Reconocimiento Combinatorio de Patrones) construyen sus conceptos en función de propiedades lógicas, basadas en los rasgos de los objetos en estudio. Además de que no requieren que sean especificados a priori el número de agrupamientos. La dificultad que poseen estos algoritmos, es la complejidad computacional en el cálculo de los testores típicos.

El operador refunión condicionada del algoritmo LC-Conceptual (ver epígrafe 1.4.1) garantiza que un concepto de un agrupamiento no sea satisfecho por ningún objeto de otro agrupamiento, pero no logra que este concepto cubra a todos los objetos del agrupamiento, por lo que en la presente investigación se opta por el uso del algoritmo RGC (ver epígrafe 1.4.2), el cual es capaz de cubrir todos los objetos de los agrupamientos formados.

1.4.1. LC-Conceptual

El algoritmo LC-Conceptual (Martínez-Trinidad y Ruiz-Shulcloper 1999) está basado en los conceptos de la clasificación no supervisada en el enfoque lógico-combinatorio (Ruiz-Shulcloper y Martínez-Trinidad 1995) y retoma algunas ideas propuestas por Michalski para generar conceptos, interpretables por los especialistas, en términos del conjunto de rasgos original.

El algoritmo LC-Conceptual (ver figura 2) consta de dos etapas: la de estructuración o determinación extensional, donde se forman los agrupamientos y la de estructuración o

determinación intencional, donde se caracteriza cada agrupamiento mediante una propiedad lógica o concepto. En la primera etapa de estructuración extensional se utilizan los conceptos del enfoque lógico combinatorio para un problema de clasificación no supervisada. En ella se construyen los agrupamientos de objetos basándose en la semejanza entre ellos y se utiliza un criterio de agrupamiento. En la segunda etapa de estructuración intencional o conceptual se construyen las propiedades (conceptos) que caracterizan a cada agrupamiento de objetos utilizando, para ello, los testores típicos y el operador de refunión condicionada (Pons-Porrata 2004).

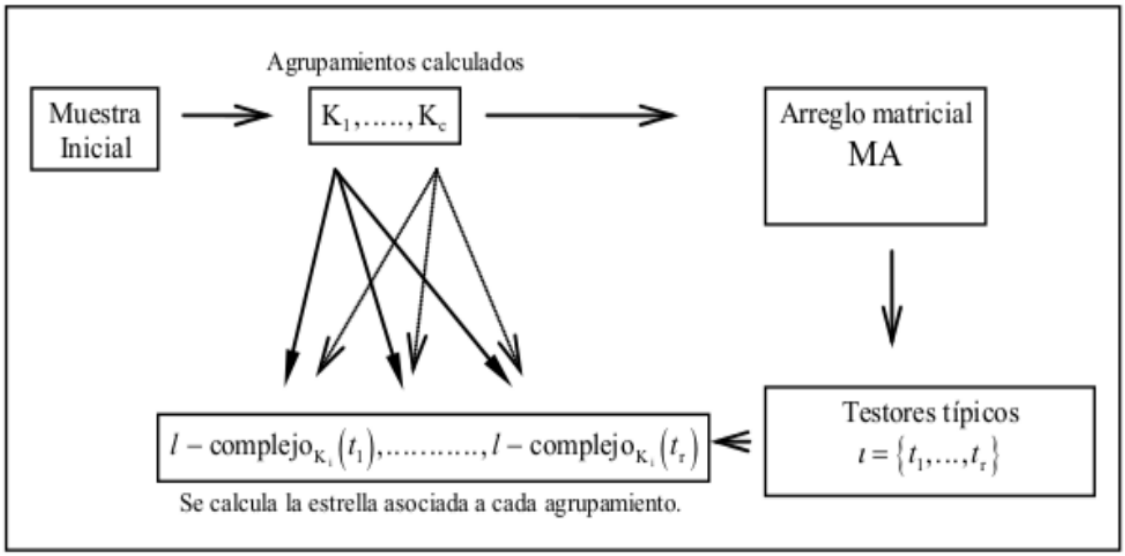


Figura 2 Proceso del agrupamiento conceptual del RGC. Fuente (Martínez-Trinidad y Sánchez-Díaz 2001)

El conjunto de testores típicos son combinaciones de variables altamente discriminantes para el conjunto de descripciones de los objetos y son estos conjuntos de variables los que se consideran para la construcción de los l-complejos. Entiéndase por l-complejo como el producto lógico de todos los rasgos que cubren un concepto o el conjunto de todos los valores diferentes que pueden tomar los objetos de dicho concepto (Pons-Porrata 2004).

El operador de refunión condicionada (RUC) transforma un conjunto de objetos y/o l-complejos en un conjunto de l-complejos determinando para cada variable el conjunto de valores que ella toma, pero no de manera independiente, sino en combinación con el resto de las variables (Pons-Porrata 2004).

1.4.2. RGC

El algoritmo RGC responde a la siguiente idea: dado un conjunto de descripciones de objetos en términos de variables simbólicas, el objetivo es encontrar una estructuración conceptual de estos objetos en el espacio de representación inicial (Pons-Porrata 2004).

Este algoritmo consta de dos etapas: una de determinación extensional y otra de determinación intencional. En la primera etapa, los agrupamientos son creados usando alguna medida de similaridad entre objetos y un criterio de agrupamiento para generar la estructuración. Luego, en la segunda etapa, los conceptos asociados a cada agrupamiento son construidos y generalizados.

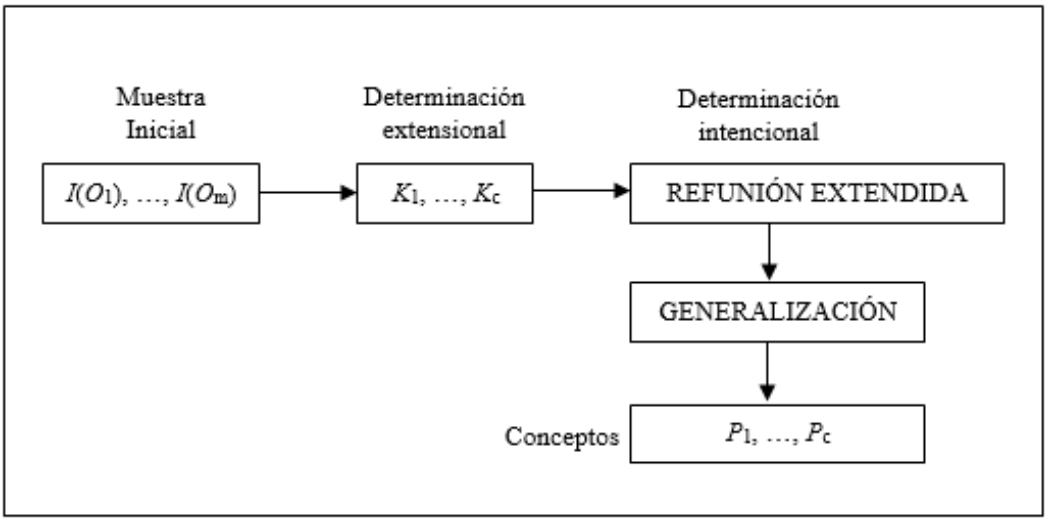


Figura 3 Proceso del agrupamiento conceptual del RGC. Fuente(Pons-Porrata 2004)

La figura 3 muestra el proceso del agrupamiento conceptual. Este algoritmo recibe su nombre, precisamente, por las tres operaciones fundamentales realizadas en la determinación intencional: refunión extendida, generalización y construcción de los conceptos.

El operador de refunión extendida (RUE) transforma un conjunto de objetos y/o l-complejos en un conjunto de l-complejos excluyentes; pues si un nuevo objeto no puede ser añadido a ninguno de los l-complejos existentes se crea, al menos, un nuevo l-complejo que lo cubre (Pons-Porrata 2004).

1.5. Análisis de herramientas existentes

Tanto a nivel nacional como internacional existen herramientas capaces de procesar información para dar paso a realizar tareas de aprendizaje automático, entre las que se considera a Weka

como la aplicación más utilizada en la actualidad a nivel internacional para la minería de datos y CEPAR es una herramienta desarrollada en Cuba, la cual tiene como fin servir de apoyo a los investigadores del Reconocimiento Lógico Combinatorio de Patrones según propone Ruiz-Shulcloper (2009).

Weka.

Es un software desarrollado por la universidad de Waikato (Nueva Zelanda), implementado en el lenguaje de programación JAVA. Contiene las herramientas necesarias para realizar transformaciones sobre los datos, tareas de clasificación, regresión, clustering, asociación y visualización. La licencia de WEKA es GPL, lo que significa que este programa es de libre distribución y difusión. Es independiente de la arquitectura (Sistema Operativo), ya que funciona en cualquier plataforma sobre la que haya una Máquina Virtual Java disponible. Está diseñado como una herramienta orientada a la extensibilidad, por lo que añadir nuevas funcionalidades es una tarea sencilla. La aplicación utiliza cubrimiento que se encuentran en documentos con extensión .arff, que son en texto plano (Pico-Peña 1995).

El software en general es bastante atractivo con una interfaz visual clara, permite hacer pruebas sobre cubrimientos, brinda distintas opciones para la visualización de los resultados y permite la ausencia de información en los datos. Se distribuye como un software de código abierto, por lo que se puede modificar cualquier elemento.

WEKA posee entre sus inconvenientes, que no permite probar iterativamente nuevos conjuntos e incluir el resultado o el nuevo conjunto clasificado al ya clasificado inicialmente, para hacer esto, hay que exportar a alguna otra aplicación y desarrollar un programa que lo haga. La metodología sobre la cual se desarrolla WEKA, no permite el tratamiento de los datos como lo hace el Reconocimiento Lógico Combinatorio de Patrones, por tanto, no es muy útil a los usuarios que hagan uso este enfoque. La incorporación de algoritmos para el Reconocimiento Lógico Combinatorio de Patrones en WEKA deben traer aparejado un cambio en el mismo núcleo de la herramienta, haciendo que pierda parte de su objetivo inicial (Pico-Peña 1995; Guzmán-Trampe 2009).

CEPAR

CEPAR (Entorno Cubano para el Reconocimiento Lógico Combinatorio de Patrones) es un Sistema Herramienta Universal (SHU) desarrollado según las teorías y presupuestos del Reconocimiento Lógico Combinatorio de Patrones. Tiene como objetivo fundamental apoyar en

las labores cotidianas de los investigadores, docentes y estudiantes de estas teorías; además de proveer de una herramienta que pueda ser embebida en desarrollos propios de una investigación. (Yero-Oses, Reyes-González y Martínez-Sánchez 2016)

Implementado empleando JAVA como lenguaje de programación, dado sus comodidades como lenguaje multiplataforma, CEPAR solo requiere de la Máquina Virtual de Java (JVM) para poder ser utilizado.

CEPAR plantea un diseño modular, de manera que permite la incorporación de nuevos rasgos, funciones de semejanza. Cada módulo es independiente y se relaciona con los demás a través de las clases e interfaces definidas en su núcleo (cepar.core), que contiene las características más generales para modelar un problema de Reconocimiento Lógico Combinatorio de Patrones.

La herramienta permite el trabajo simultáneo con variables cualitativas y cuantitativas (todos los tipos de rasgos soportan la ausencia de información o valor “?”), y maneja de forma nativa los tipos:

- BooleanFeature: valores booleanos.

- DateFeature: valores de tipo fecha (día-mes-año).

- DoubleFeature: valores reales.

- FloatFeature: valores de punto flotante.

- IntegerFeature: valores enteros.

- StringFeature: valores alfanuméricos.

Esta cantidad de tipos de rasgos resulta mínima, en comparación con los que pueden aparecer en problemas reales del Reconocimiento Lógico Combinatorio de Patrones, por lo que resalta en CEPAR el permitir la extensión según las interfaces definidas de nuevos rasgos por los usuarios, pero limitando la herramienta sólo al trabajo con los seis rasgos nativos.

Por las potencialidades del uso del enfoque del Reconocimiento Lógico Combinatorio de Patrones que brinda la herramienta CEPAR y que es capaz de insertar nuevos conjuntos de datos clasificados a los ya clasificados inicialmente, en la presente investigación se utiliza la herramienta como librería para establecer las bases en la implementación del algoritmo conceptual RGC.

1.6. Lenguajes de programación

Los lenguajes de programación facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por las personas y a su vez resultan independientes del modelo de computador a utilizar. Para la implementación del algoritmo conceptual RGC se propone la utilización de java en su versión 8 y para la realización de las pruebas estadísticas se propone el uso de R.

Java 8

Versión más reciente de Java que incluye nuevas características, mejoras y correcciones de bugs para mejorar la eficacia en el desarrollo y la ejecución de programas Java. A continuación, se muestra un breve resumen de las mejoras que se incluyen en esta versión:

- Detectar y eliminar versiones de Java antiguas (Windows) a partir de Java 8 Update 20 (8u20), en los sistemas Windows, la herramienta de desinstalación de Java está integrada con el installer para contar con una opción para eliminar las versiones anteriores de Java del sistema. El cambio se aplica a plataformas Windows de 32 bits y 64 bits.
- Métodos de extensión virtual y expresión Lambda: una de las funciones destacables de Java SE 8 es la implantación de expresiones Lambda y funciones adyacentes a la plataforma y el lenguaje de programación Java.
- API de fecha y hora: esta nueva API permitirá a los administradores gestionar datos de fecha y hora de forma mucho más natural y fácil de comprender.
- Motor de JavaScript Nashhorn: nueva implantación ligera de alto rendimiento del motor de JavaScript integrada en JDk y disponible en las aplicaciones Java mediante las API existentes.
- Seguridad mejorada: sustitución de la lista de métodos sensibles al emisor mantenida a mano existente por un mecanismo que identifica con mayor precisión dichos métodos y permite detectar a los emisores de forma fiable.

R

R es un lenguaje y un entorno para computación y gráficos estadísticos que proporciona una amplia variedad de técnicas estadísticas y gráficos, y es altamente extensible. El lenguaje R suele ser el vehículo de elección para la investigación en metodología estadística y proporciona una ruta de código abierto para la participación en esa actividad («R: What is R?» 2018).

Uno de los puntos fuertes de R es la facilidad con la que se pueden producir parcelas de calidad de publicación bien diseñadas, que incluyen símbolos matemáticos y fórmulas cuando es necesario. Se ha tenido mucho cuidado con los valores predeterminados para las opciones menores de diseño en los gráficos, pero el usuario conserva el control total. R está disponible como software libre bajo los términos de la Licencia Pública General GNU de la Free Software Foundation en forma de código fuente. Se compila y se ejecuta en una amplia variedad de plataformas UNIX y sistemas similares (incluidos FreeBSD y Linux), Windows y MacOS («R: What is R?» 2018).

R permite a los usuarios agregar funciones adicionales definiendo nuevas funciones. Gran parte del sistema está escrito en el dialecto R, lo que facilita a los usuarios seguir las elecciones algorítmicas realizadas. Para tareas intensivas en cómputo, el código C, C ++ y Fortran se puede vincular y ejecutar en tiempo de ejecución. Los usuarios avanzados pueden escribir código C para manipular objetos R directamente. («R: The R Project for Statistical Computing» 2018)

1.7. Entornos de desarrollo integrado

Un Entorno de Desarrollo Integrado (IDE), es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Como entorno de desarrollo integrado se utilizará **NetBeans** en su versión 8.2 para la implementación del algoritmo conceptual y RStudio con el objetivo de realizar las pruebas no paramétricas de Friedman.

NetBeans

Está escrito en Java, pero puede servir para lenguajes de programación como C++, HTML5, JavaScript y otros. Existe además un número importante de módulos para extender el Netbeans IDE. Netbeans IDE es un producto libre y gratuito sin restricciones de uso (Oracle Corporation 2018).

A continuación, algunas ventajas de este IDE que se consideran importantes en la investigación:

- Es un IDE multilenguaje y adaptable.
- Es software libre y gratuito.
- Intuitivo y fácil de utilizar.
- Se puede desarrollar todo tipo de aplicaciones.
- Es poderoso y extensible.

RStudio

Basado en el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo. Es multiplataforma para los sistemas operativos Windows, Mac y Linux o para navegadores conectados a RStudio Server o RStudio Server Pro. RStudio tiene la misión de proporcionar el entorno informático estadístico R. Permite un análisis y desarrollo para que cualquiera pueda analizar los datos con R (Santana y Nieves-Hernándes 2018).

1.8. Frameworks

La palabra *framework* define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. Los objetivos principales que persigue un *framework* son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

jCOLIBRI

Desarrollada por el Grupo de Aplicaciones de Inteligencia Artificial (GAIA) de la Universidad Complutense de Madrid. Esta plataforma de software permite crear aplicaciones utilizando RBC de forma rápida y sencilla, siendo utilizada en la actualidad como una herramienta educativa y de desarrollo(«GAIA – Group of Artificial Intelligence Applications | Universidad Complutense de Madrid, Spain» 2011).

Su arquitectura se divide en dos capas que cubren las distintas necesidades tanto de sus usuarios desarrolladores como diseñadores. Los usuarios desarrolladores son aquellos que prefieren manejar directamente el código de la aplicación, mientras que los diseñadores prefieren utilizar herramientas de composición a más alto nivel.

La parte de la plataforma orientada a usuarios desarrolladores ofrece todos los elementos básicos que sirven de base en la implementación de sistemas usando RBC. La otra capa de jCOLIBRI se basa en las tecnologías de la Web Semántica para permitir la composición semiautomática de las aplicaciones RBC por parte de usuarios diseñadores. Gracias a los nuevos estándares propuestos en esta área es posible representar y razonar sobre el comportamiento de los métodos del RBC

incluidos en la plataforma y así asistir al usuario en su composición.(Recio-García, González-Calero y Díaz-Agudo 2014)

myCBR

myCBR es una herramienta de recuperación basada en similitudes de código abierto y un kit de desarrollo de software (SDK). Con myCBR Workbench puede modelar y probar medidas de similitud altamente sofisticadas e intensivas en conocimiento en una poderosa GUI e integrarlas fácilmente en sus propias aplicaciones utilizando myCBR SDK. Los sistemas de recomendación de productos basados en casos son solo un ejemplo de aplicaciones de recuperación basadas en similitudes («myCBR» 2006).

MyCBR Workbench es una potente GUI capaz de modelar medidas de similitud intensivas en conocimientos. Es utilizado en un entorno de desarrollo integrado con Eclipse. Proporciona configuraciones orientadas a tareas para modelar su conocimiento, extracción de información y manejo de la base de casos; consta de funcionalidades dedicadas a la recuperación de casos similares y permite la representación de casos como objetos estructurados.

Los *frameworks* descritos anteriormente no son usados en la propuesta de solución, ya que tienen la peculiaridad de trabajar con datos solo de tipo texto, lo que complejiza el uso de los mismo en el Reconocimiento Lógico Combinatorio de Patrones, pues este vincula datos cualitativos y cuantitativos.

1.9. Conclusiones parciales

Como resultado del análisis del estado del arte para determinar los referentes teóricos más importantes y actuales sobre las estructuras de organización de la base de casos en los sistemas basados en casos, se pudo constatar que:

- Existen deficiencias en las estructuras de organización de la base de casos, donde las jerarquías constituyen el principal modelo de representación. En particular, las limitaciones en las estructuras jerárquicas conceptuales evidencian la necesidad de continuar trabajando en el desarrollo de nuevas formas de organización de la base de casos.

El uso del algoritmo conceptual RGC permite realizar una conceptualización capaz de cubrir todos los objetos y que no existan objetos semejantes en otros agrupamientos formados, lo cual puede ser aprovechado en la construcción de la base de conocimientos de un sistema basado en casos.

Capítulo 2. Estructura de organización de la base de casos utilizando el algoritmo de agrupamiento conceptual RGC

El capítulo describe la propuesta de solución del Sistema Basado en Casos en el que se sustenta el algoritmo conceptual RGC. Se realiza una descripción detallada de cada una de las etapas de dicho algoritmo, así como de la estructura jerárquica conceptual de la base de casos propuesta en conjunto con dos algoritmos donde se realiza el agrupamiento conceptual y la clasificación de nuevos objetos en la base de casos.

2.1. Descripción de la propuesta de solución

La figura 4 muestra la propuesta de solución, tomando como referencia los elementos del modelo propuesto por Reyes-González (2017) el cual está constituido por dos componentes: representación del conocimiento y soporte a las decisiones. En el primer componente se realizan todos los procesos relacionados con las etapas de determinación extensional e intencional del algoritmo conceptual RGC. Luego de construida la estructura jerárquica para la base de casos, el segundo componente al recibir un nuevo caso en la base de conocimiento realiza el proceso de acceso y recuperación de los casos más similares determinando una posible solución.

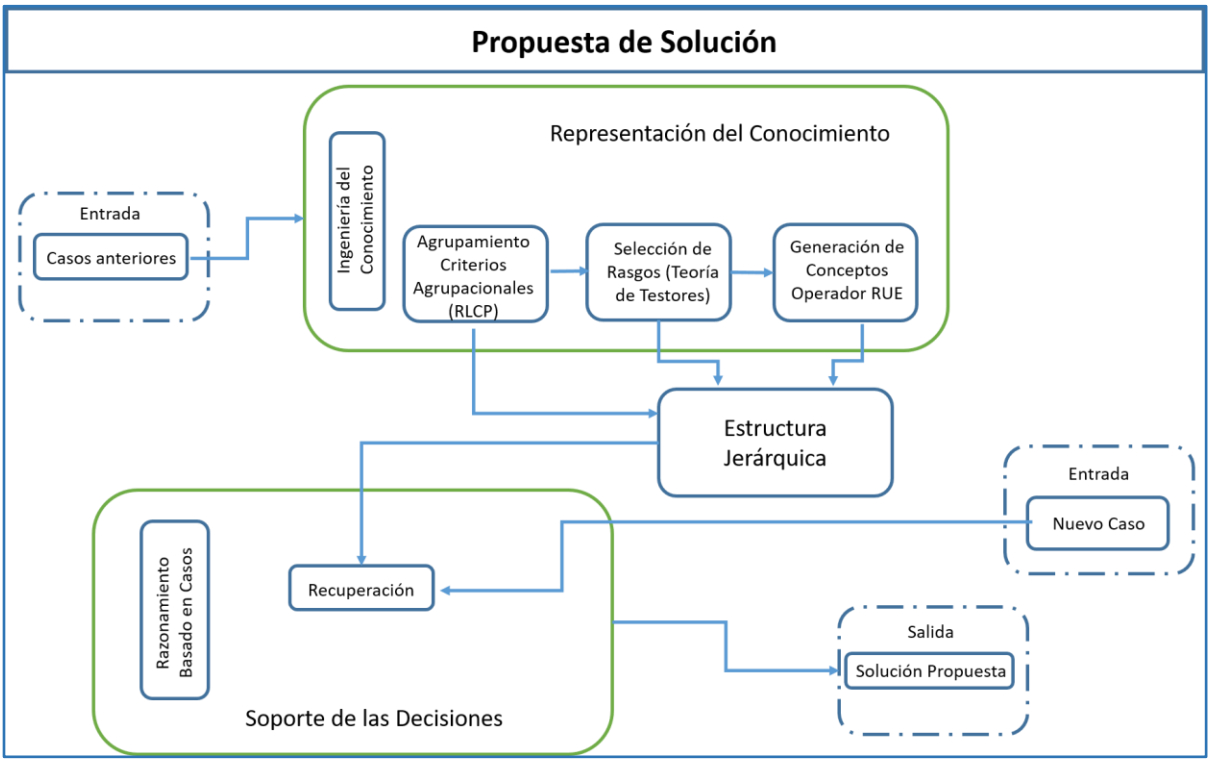


Figura 4 Representación gráfica de la propuesta de solución. Fuente (Elaboración propia)

El punto de partida de la propuesta de solución es la utilizando el algoritmo conceptual RGC, el cual permite realizar un análisis minucioso sobre las características de los datos, su significado y se decide la manera en que se comparan los rasgos y los casos.

Este algoritmo tiene como entradas el conjunto de casos resueltos con anterioridad, los cuales son agrupados para obtener conglomerados de casos similares. Se determinan los testores típicos, así como el peso informacional de los rasgos que los conforman y se selecciona el testor típico de mayor relevancia con el que se construyen los conceptos distintivos para cada grupo.

Los procesos del segundo componente, se describen a través del ciclo del razonamiento basado en casos que se inicia al presentarse un problema a resolver. Se aplica el algoritmo para el acceso al grupo donde se encuentran los casos más similares y se recuperan aquellos que se deben tener en cuenta para la reutilización. Para conformar la propuesta de solución se propone utilizar la del caso más similar.

2.2. Representación de la base de casos mediante una estructura jerárquica conceptual

Para realizar este proceso se propone adoptar la metodología descrita por Ruiz-Shulcloper (2009) la cual precisa de varios aspectos a tener en cuenta, tales como: cuáles son los rasgos y su importancia, cómo se comparan las variables y los objetos que están agrupados en clases duras y disjuntas. Los casos constituyen el objeto de estudio en los Sistemas Basados en Casos. Están conformados por los rasgos predictores $(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n})$ y el rasgo objetivo (S) y pueden definirse de la siguiente manera en la ecuación 2:

$$c = \{(x_1, x_{2,\ldots}, x_n), (s)\} \qquad (2)$$

Cada rasgo constituye una variable aleatoria cualitativa o cuantitativa con la que se describen los casos. En dependencia de la naturaleza del rasgo pueden utilizarse diferentes criterios de comparación tales como los mostrados en las funciones de comparación siguientes o pueden construirse nuevas funciones que no necesariamente sean booleanas.

Criterio de comparación igualdad estricta (ver ecuación 3)

$$\delta_s\left(x_s(o_i), x_s(o_j)\right) = \begin{cases} 1\ si\ x_s(o_i) = x_s(o_j) \vee x_s(o_i) = * \vee x_s(o_j) = * \\ 0\ en\ otro\ caso \end{cases} \qquad (3)$$

Intervalos de valores semejantes (ver ecuación 4)

$$\delta_s\left(x_s(o_i), x_s(o_j)\right) = \begin{cases} 1\ si\ x_s(o_i), x_s(o_j)\ \in [a_p, a_{p+1}\,] \vee x_s(o_i) = * \vee\ x_s(o_j) = * \\ 0\ en\ otro\ caso \end{cases} \quad (4)$$

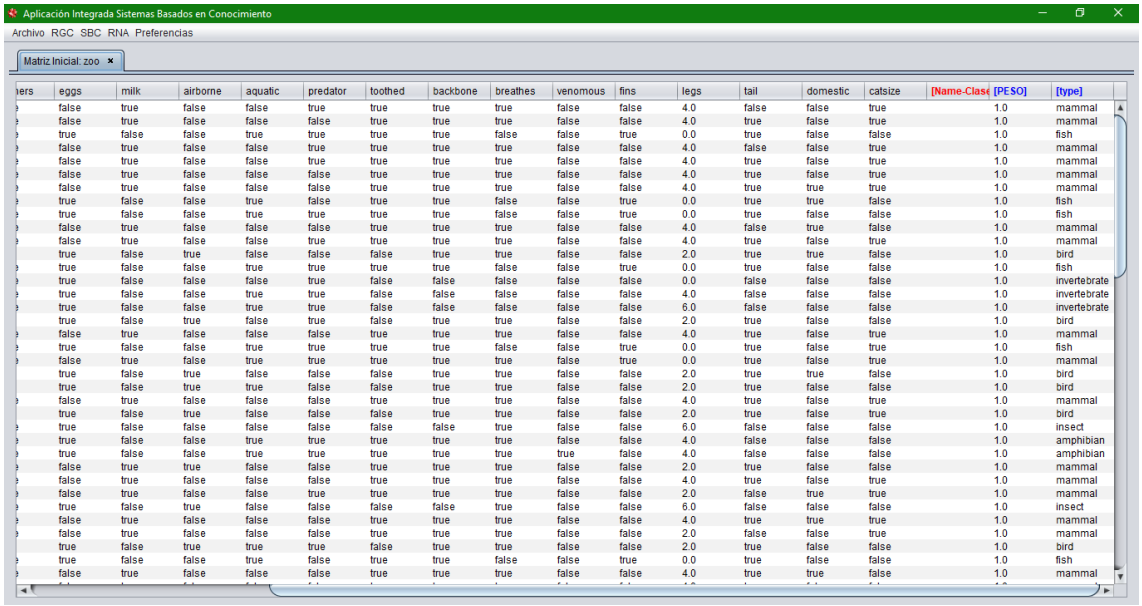
Umbral de error admisible de semejanza (ver ecuación 5)

$$\delta_s\left(x_s(o_i), x_s(o_j)\right) = \begin{cases} 1\ si\ x_s(o_i), x_s(o_j) < \varepsilon \vee x_s(o_i) = * \vee x_s(o_j) = * \\ 0\ en\ otro\ caso \end{cases} \quad (5)$$

Conjunto de valores semejantes (ver ecuación 6)

$$\delta_s\left(x_s(o_i), x_s(o_j)\right) = \begin{cases} 1\ si\ x_s(o_i), x_s(o_j) \in A_p, \vee x_s(o_i) = * \vee x_s(o_j) = * \\ 0\ en\ otro\ caso \end{cases} \quad (6)$$

La base de conocimientos se construye a partir de los casos existentes con su estructura establecida. Se representa como una matriz inicial (MI, ver figura 5), donde *n* es la cantidad de rasgos (características que describen los casos) conforman las columnas y *m* el número de casos que se corresponden con las filas. Se consideran sólo los rasgos predictores para obtener una estructuración de la base de casos, tratándose como un problema no supervisado.



ers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	legs	tail	domestic	catsize	[Name-Clas	[PESO]	[type]
	false	true	false	false	true	true	true	true	false	false	4.0	false	false	true		1.0	mammal
	false	true	false	false	false	true	true	true	false	false	4.0	true	false	true		1.0	mammal
	true	false	false	true	true	true	true	false	false	true	0.0	true	false	false		1.0	fish
	false	true	false	false	true	true	true	true	false	false	4.0	false	false	true		1.0	mammal
	false	true	false	false	true	true	true	true	false	false	4.0	true	false	true		1.0	mammal
	false	true	false	false	false	true	true	true	false	false	4.0	true	false	true		1.0	mammal
	false	true	false	false	false	true	true	true	false	false	4.0	true	true	true		1.0	mammal
	true	false	false	true	false	true	true	false	false	true	0.0	true	true	false		1.0	fish
	true	false	false	true	true	true	true	false	false	true	0.0	true	false	false		1.0	fish
	false	true	false	false	false	true	true	true	false	false	4.0	false	true	false		1.0	mammal
	false	true	false	false	false	true	true	true	false	false	4.0	true	false	true		1.0	mammal
	true	false	true	false	false	false	true	true	false	false	2.0	true	true	false		1.0	bird
	true	false	false	true	true	true	true	false	false	true	0.0	true	false	false		1.0	fish
	true	false	false	false	true	false	false	false	false	false	0.0	false	false	false		1.0	invertebrate
	true	false	false	true	true	false	false	false	false	false	4.0	false	false	false		1.0	invertebrate
	true	false	false	true	true	false	false	false	false	false	6.0	false	false	false		1.0	invertebrate
	true	false	true	false	true	false	true	true	false	false	2.0	true	false	false		1.0	bird
	false	true	false	false	false	true	true	true	false	false	4.0	true	false	true		1.0	mammal
	true	false	false	true	true	true	true	false	false	true	0.0	true	false	true		1.0	fish
	false	true	false	true	true	true	true	true	false	true	0.0	true	false	true		1.0	mammal
	true	false	true	false	false	false	true	true	false	false	2.0	true	true	false		1.0	bird
	true	false	true	true	false	false	true	true	false	false	2.0	true	false	false		1.0	bird
	false	true	false	false	false	true	true	true	false	false	4.0	true	false	true		1.0	mammal
	true	false	true	false	false	false	true	true	false	false	2.0	true	false	true		1.0	bird
	true	false	false	false	false	false	false	true	false	false	6.0	false	false	false		1.0	insect
	true	false	false	true	true	true	true	true	false	false	4.0	false	false	false		1.0	amphibian
	true	false	false	true	true	true	true	true	true	false	4.0	false	false	false		1.0	amphibian
	false	true	true	false	false	true	true	true	false	false	2.0	true	false	false		1.0	mammal
	false	true	false	false	false	true	true	true	false	false	4.0	true	false	true		1.0	mammal
	false	true	false	false	true	true	true	true	false	false	2.0	false	true	true		1.0	mammal
	true	false	true	false	false	false	true	true	false	false	6.0	false	false	false		1.0	insect
	false	true	false	false	false	true	true	true	false	false	4.0	true	true	true		1.0	mammal
	false	true	false	false	false	true	true	true	false	false	2.0	false	false	true		1.0	mammal
	true	false	true	true	true	false	true	true	false	false	2.0	true	false	false		1.0	bird
	true	false	false	true	false	true	true	false	false	true	0.0	true	false	false		1.0	fish
	false	true	false	false	true	true	true	true	false	false	4.0	true	true	false		1.0	mammal

Figura 5 Matriz Inicial. Fuente (Elaboración propia)

Para una mayor comprensión del algoritmo RGC es necesario conocer los procesos que realiza en cada una de sus etapas, las cuales se describen a continuación:

Etapa extensional:

- ✓ Se seleccionan los criterios de comparación de valores C_i para cada variable x_i, $i = 1, \ldots, n$ la función de semejanza entre objetos Γ, el criterio agrupacional Π
- ✓ Se obtienen los agrupamientos (duros) $K1, \ldots, Kc$ de *MI*, aplicando Π.
- ✓ Determinación del conjunto de apoyo.

La función de semejanza determina una medida numérica del grado de similitud de un caso con respecto al otro teniendo en cuenta las similitudes entre los rasgos. Pueden utilizarse diferentes funciones de semejanza, de acuerdo al criterio de los especialistas del problema en particular, no obstante, se sugiere utilizar por defecto la ecuación de la suma pesada de las funciones de comparación por rasgos como se muestra en el primer paso del algoritmo.

Para calcular el umbral de semejanza se sugieren varias alternativas como se ilustra en la figura 6.

Figura 6 Variantes para determinar el umbral. Fuente (Elaboración propia)

La selección del criterio de agrupamiento se deja abierta según las características del dominio de aplicación y la valoración de los expertos, pudiendo decidirse entre aquel que calcula las componentes βo-Conexas o los conjuntos βo-Compactos (RuizShulcloper 2009). Debe tenerse en cuenta que en el caso de los conjuntos compactos estos son más restrictivos en cuanto a la formación de los agrupamientos porque los construye con los objetos de máxima semejanza, y por tanto al utilizar este criterio se obtiene un mayor número de conjuntos que con el conexo. Se recomienda utilizar el criterio βo-Conexo y si es necesario volver a aplicar dentro de algún agrupamiento otro criterio entonces se sugiere el βo-Compacto, aunque esta elección depende del problema en particular (ver figura 7).

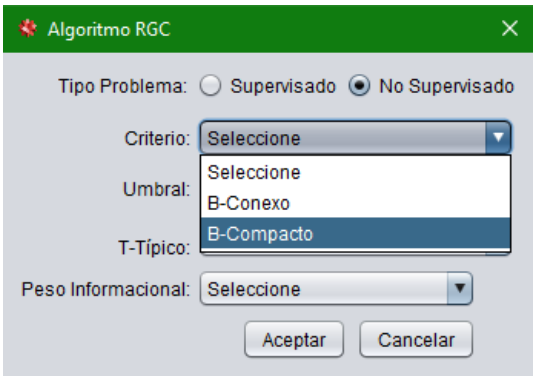


Figura 7 Criterios de agrupamiento. Fuente (Elaboración propia)

Una vez que se obtienen los agrupamientos es necesario determinar el conjunto de apoyo el cual se forma utilizando la teoría de testores como herramienta matemática para la selección de rasgos. De este modo se reduce la cantidad de atributos con los cuales se deben describir los casos y se determinan los que inciden de manera determinante en el problema.

Según el experimento realizado por Reyes-González (2017) muestra que al aplicar el algoritmo Fast-BR se logra una mayor reducción de los testores típicos con respecto al método AttributeSelection existente en Weka, por lo que se recomienda el uso del algoritmo Fast-BR por defecto para realizar el cálculo de los testores típicos (ver figura 8).

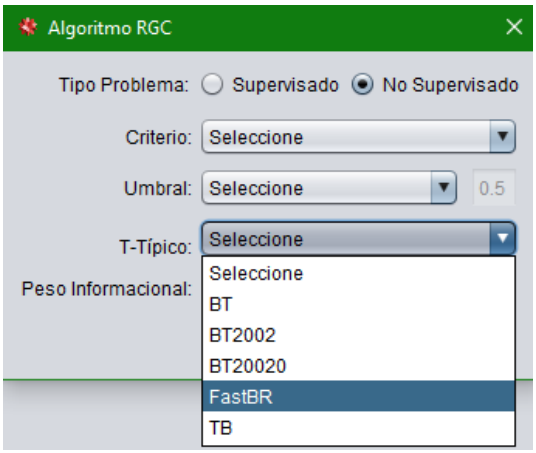


Figura 8 Algoritmos para el caculo de los Testores Típicos. Fuente (Elaboración propia)

Etapa Intencional:

- ✓ Se forma una matriz de aprendizaje *MA* a partir de las subdescripciones de *MI* según τ y de los agrupamientos $K_1, \ldots, K_c$.

- ✓ Calcular la estrella $G_t(\frac{K_i}{K_1}, \ldots, K_i - 1, K_i + 1, \ldots, K_c)$. Para el cálculo de los $\mathrm{R}|_{\mathrm{K_i}}$ en cada $X \in \tau$ se emplea el operador de refunión extendida *RUE*.
- ✓ Aplicar el operador de generalización *GEN* a todas las variables de cada l-complejo obtenido en el paso anterior y construir los l-complejos generalizados $\alpha_1, \ldots, \alpha_q$.
- ✓ Determinar los objetos $O_1, \ldots, O_t \in \frac{M_i}{K_i}$ que satisfacen a la propiedad $\alpha_1 \vee \ldots \vee \alpha_q$.
- ✓ El concepto que caracteriza al agrupamiento K_i será: $P_i = (\alpha_j \vee \ldots \vee \alpha_q) \wedge \neg(\alpha O_j \vee \ldots \vee \alpha O_t)$, donde αO_j , $j = 1, \ldots, t$ son los l-complejos elementales asociados a los objetos $O_1, \ldots, O_t$.

El concepto de cada uno de los agrupamientos es determinado a partir de la MA que es formada con el apoyo de los testores típicos de mayor peso informacional y la MI de cada problema, luego se aplica el procedimiento del cálculo de la estrella que emplea el operador RUE (Pons-Porrata 2004) para obtener los conceptos expresados en l-complejos (ver figura 9-10).

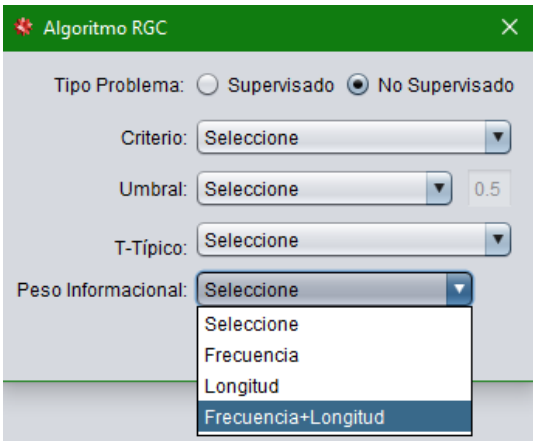


Figura 9 Peso informacional. Fuente (Elaboración propia)

En el cálculo de la estrella para cada l-complejo de cada clase K_i se generan todas las combinaciones posibles con los valores que toman los rasgos que los conforman y se seleccionan aquellos l-complejos que sólo representan a objetos de la clase K_i, siendo estos excluyentes y caracterizantes.

Los l-complejos son considerados excluyentes cuando no existe ningún objeto fuera del agrupamiento K_i que esté cubierto por los conceptos generados de dicho agrupamiento y se consideran caracterizantes cuando el concepto generado cubre a todos los objetos existentes en K_i.

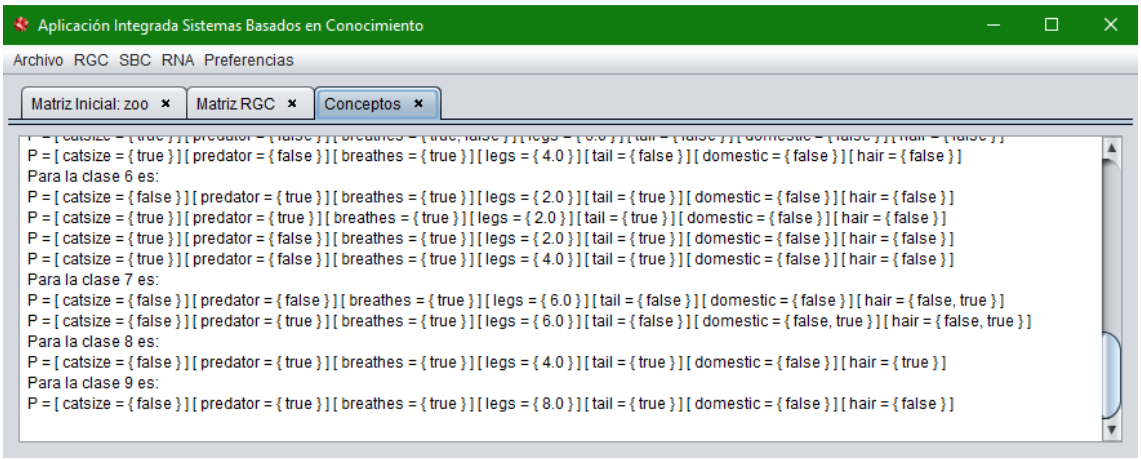


Figura 10 Conceptos generados. Fuente (Elaboración propia)

Las reglas de generalización, se aplican como una mejora para la simplificación y generalización de los valores de los rasgos en los conceptos, en correspondencia con el tipo de variable. Se tiene siempre en cuenta, al aplicar una regla de generalización, que el concepto generalizado no cubra más objetos que los que cubría sin ser generalizado. Los rasgos booleanos y cualitativos nominales no requieren de la aplicación de reglas de generalización. A continuación, se muestra un ejemplo de estas reglas para variables cuantitativas y cualitativas basadas en Martínez-Trinidad, Ruiz-Shulcloper y Pons-Porrata (1999):

<u>Regla de generalización para rasgos de tipo cuantitativos y cualitativos ordinales</u>

1. C´= cantidad de objetos de MI\K cubiertos por los valores de R_i.
2. Tomar el intervalo más pequeño que cubra a los valores en R_i
3. Determinar la cantidad C de objetos de MI\K cubiertos por el intervalo.
4. Si C=C' entonces R_i' es el intervalo.

   Si no

   Hallar el conjunto de los k subintervalos disjuntos I_j formados a partir de los valores de R_i tal $\sum_{j=1}^{K} c_j$ donde C_j es la cantidad de objetos cubiertos por el intervalo I_J.

   $R_i' = \{I_J\};\ j = 1 \ldots k.$

Una vez explicadas cada una de las etapas con las que cuenta el algoritmo RGC se propone el siguiente algoritmo:

Algoritmo 1: Agrupamiento conceptual.

Entradas:

- ✓ Una colección de objetos MI a ser agrupados.
- ✓ Criterios de comparación Ci para cada variable X_i, $i = 1, \dots, n$.
- ✓ Una función de semejanza entre objetos Γ.
- ✓ Un criterio agrupacional Π y los parámetros que éste requiera.

Salida: una estructuración (dura) no necesariamente disjunta de *MI* con los conceptos que caracterizan a cada uno de estos agrupamientos.

Paso 1: Agrupamiento de los casos

Entrada: MI / / Matriz inicial

β / / Función de semejanza

π / / Criterio de agrupamiento

C_m / / Casos, m recorre la cantidad de casos

$\delta\left(x_i(o_j), x_n(o_m)\right)$ / / Funciones de comparación por rasgos

Salida: ME // Matriz estructurada en c agrupamiento K_c

1: Construir la matriz de semejanza utilizando la función de semejanza (ecuación 7)

$$\beta(C_1, C_m) = \frac{\sum_{i=1}^{n} \delta(x_i(C_i), x_n(C_m))}{n} \qquad (7)$$

2: Calcular el umbral de semejanza utilizando un criterio β_0 según ecuación 8 o 9.

$$\beta_0 = \frac{i}{m}\left\{\sum_{i=1}^{m} \underset{i \neq j=1,\dots,m}{max}\left\{\beta\left(I(C_i), I(C_j)\right)\right\}\right\} \qquad (8)$$

$$\beta_0 = \underset{\substack{i=l_m-1 \\ i \neq j}}{min}\left\{\underset{j=i+l\_m}{min}\{\beta\left(I(O_i), I(O_j)\right)\}\right\} \qquad (9)$$

3: Aplicar criterio de agrupamiento β_0-Conexo o β_0-Compacto

Paso 2: Cálculo de los testores típicos.

Entrada: ME / / Matriz de Entrenamiento

α y γ / / Parámetros para ponderar la frecuencia y longitud

Salida: tt / / Testor Típico de mayor peso informacional

ε_i / / Importancia de los rasgos que aparecen en los testores típicos

1: Calcular la matriz de diferencias.

2: Calcular matriz básica.

3: Aplicar algoritmo para el cálculo de los testores típicos.

4: Calcular el peso ε_i de los rasgos x_i que aparecen en la familia de testores típicos.

5: Seleccionar el testor típico de mayor peso informacional (tt).

Paso 3: Construcción de los Conceptos.

Entrada: ME / / Matriz de entrenamiento

tt / / Testor típico de mayor peso informacional

Salida: l-complejos / / Complejos lógicos de cada agrupamiento

1: Calcular la estrella $G_t(\frac{K_i}{K_1}, \ldots, K_i - 1, K_i + 1, \ldots, K_c)$ para cada clase K_i i = 1 ,…, c utilizando el operador de Refunión Extendida(RUE).

2: Aplicar reglas de generalización en dependencia del tipo de variable.

Los conceptos son construidos a partir de la unión de los l-complejos formados por cada agrupamiento K_c.

En el paso 2 primero se calcula la Matriz de Diferencias (MD) y luego la Matriz Básica (MB), a partir de la cual se determinan los testores.

La Matriz de Diferencias (ver figura 11) es una matriz booleana que se obtiene de la matriz de aprendizaje comparando los respectivos valores de los rasgos en objetos de clases diferentes por medio de los criterios de comparación de valores de las variables.

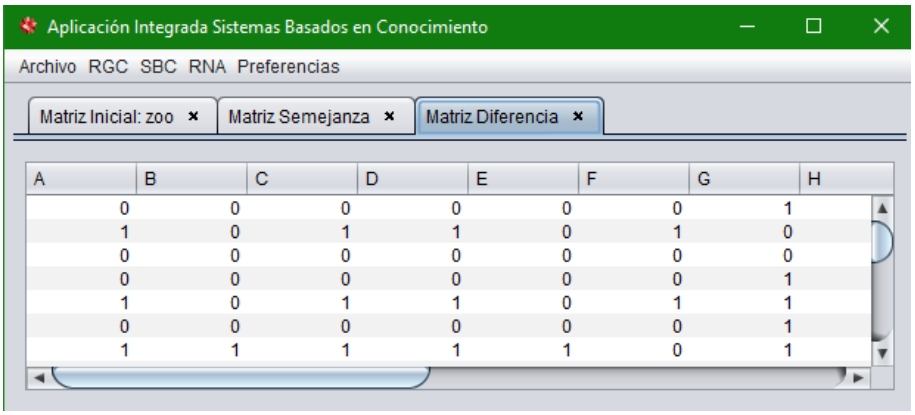


Figura 11 Matriz de Diferencias. Fuente (Elaboración propia)

Dada una Matriz de Diferencias se le llama Matriz Básica (ver figura 12) a la matriz formada exclusivamente por filas básicas de MD.

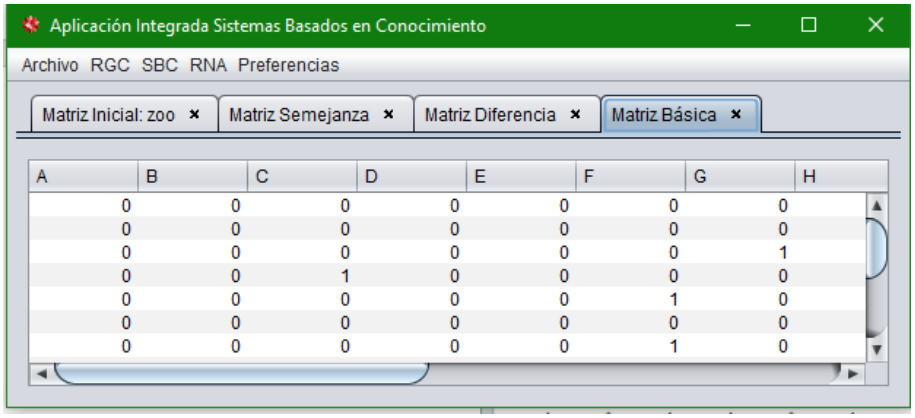


Figura 12 Matriz Básica. Fuente (Elaboración propia)

Fila básica (Ruiz-Shulcloper 2009)

La fila i_t es básica sí y sólo sí en *MD* no existe fila i_p alguna que sea subfila de i_t.

Subfila

Sea i_p, i_t filas de *MD* Se dice que i_p es subfila de i_t sí y sólo sí:

a) $\forall j \; (a_{i_t j}=0 \Rightarrow a_{i_p j}=0)$

b) $\exists j_0 \; (a_{i_t j_0}=1 \wedge a_{i_p j_0}=0)$

Las filas de la matriz básica constituyen el conjunto de testores.

2.3. Estructuración jerárquica conceptual de la base de casos

La base de casos responde a una estructura jerárquica conceptual, en la que se destacan en los niveles superiores de la jerarquía los conceptos que se corresponden con los agrupamientos. El nodo raíz contiene una abstracción de todos los casos del sistema. En el segundo nivel se localizan los conceptos expresados mediante complejos lógicos.

Ejemplo 1: Un concepto P generado a partir del TT = {r_0, r_3} con valores del rasgo 1 y rasgo 4 donde v_1=v_2={bueno, malo} presentes en los objetos de la clase a la cual P representa, puede expresarse de la forma:

$$P=\{(r_0=[\text{ bueno, malo}]\wedge r_3=[\text{ malo}]) \vee\{(\ r_0=[\text{ malo}]\wedge r_3=[\text{ bueno, malo}])\}$$

Estos conceptos generados cubren objetos observados y no observados (Michalski 1979). Los observados son aquellos que representan objetos cuyas descripciones aparecen en el agrupamiento y los no observados son combinaciones de valores, generadas como resultado de aplicar el operador RUE, que se corresponden con objetos que no están físicamente en el grupo (Reyes-González 2017).

La figura 11 muestra el diseño de una base de casos con una estructurada jerárquica conceptual según (Reyes-González 2017), donde P corresponde al l-complejo de cada uno de los n grupos o clases generadas. En dependencia del número de casos en cada clase se puede volver a aplicar dicho procedimiento, para obtener nuevos niveles de jerarquía.

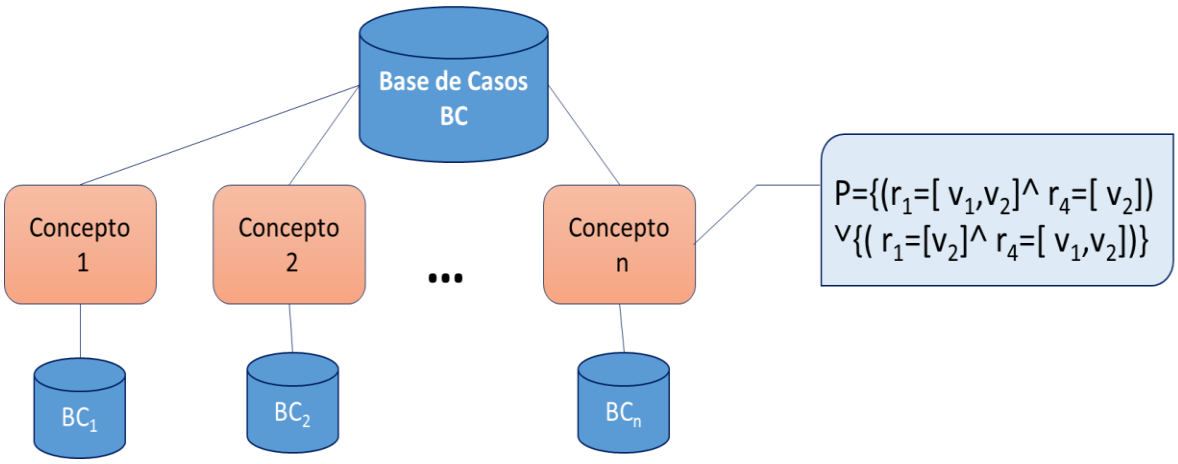


Figura 13 Representación de la base de casos en jerarquía conceptual. Fuente (Reyes-González 2017)

Para el acceso y recuperación de los casos semejantes la base de casos debe poseer una estructura que facilite este proceso. El acceso eficaz se garantiza a través de los conceptos representativos para cada agrupamiento y la recuperación de los casos semejantes se realiza a través de la búsqueda hacia el interior de estos. Teniendo en cuenta a (Reyes-González 2017), plantea que es necesario determinar los subconceptos de cada conjunto obtenidas a partir de la descomposición de los conceptos en subconjuntos de todas las posibles combinaciones de valores de los diferentes rasgos que representan a los objetos del agrupamiento (ver Ejemplo 2).

Ejemplo 2: Del concepto P_1, se derivan los subconceptos:

$$b_1 = \{r_1v_1,-, -, r_4v_2,-\},\ b_2 = \{r_1v_2,-, -, r_4v_1,-\},\ b_3 = \{r_1v_2,-, -, r_4v_2,-\}$$

Con relación a los subconceptos que representan objetos no observados como: $\{r_1v_2,-, -, r_4v_2,-\}$, puede afirmarse que estos son casos que no existen físicamente en la base de conocimientos pero que son posibles en el dominio de aplicación, por lo que deben ser valorados por los expertos para tomar una decisión respecto a su aparición como problema real y proponer una solución.

Hacia el interior de cada agrupamiento los subconceptos se organizan en forma de árbol jerárquico para facilitar la recuperación de los casos semejantes. Se debe tener en cuenta que la búsqueda en esta estructura se inicia realizando recorridos de izquierda a derecha (ver figura 12).

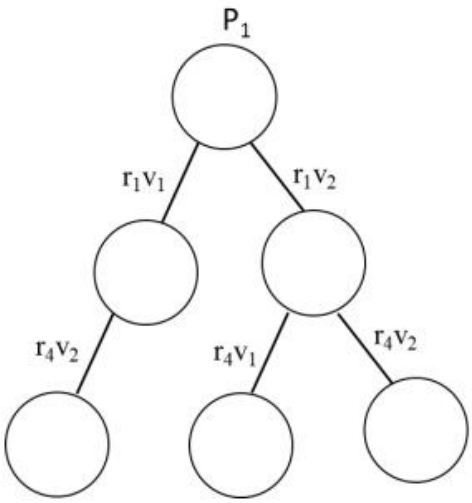


Figura 14 Representación jerárquica. Fuente (Reyes-González 2017)

Una vez conformada la estructura jerárquica conceptual de la base de conocimiento se propone el algoritmo 2 para realizar la clasificación de nuevos objetos (ver figura 15) en la base de casos:

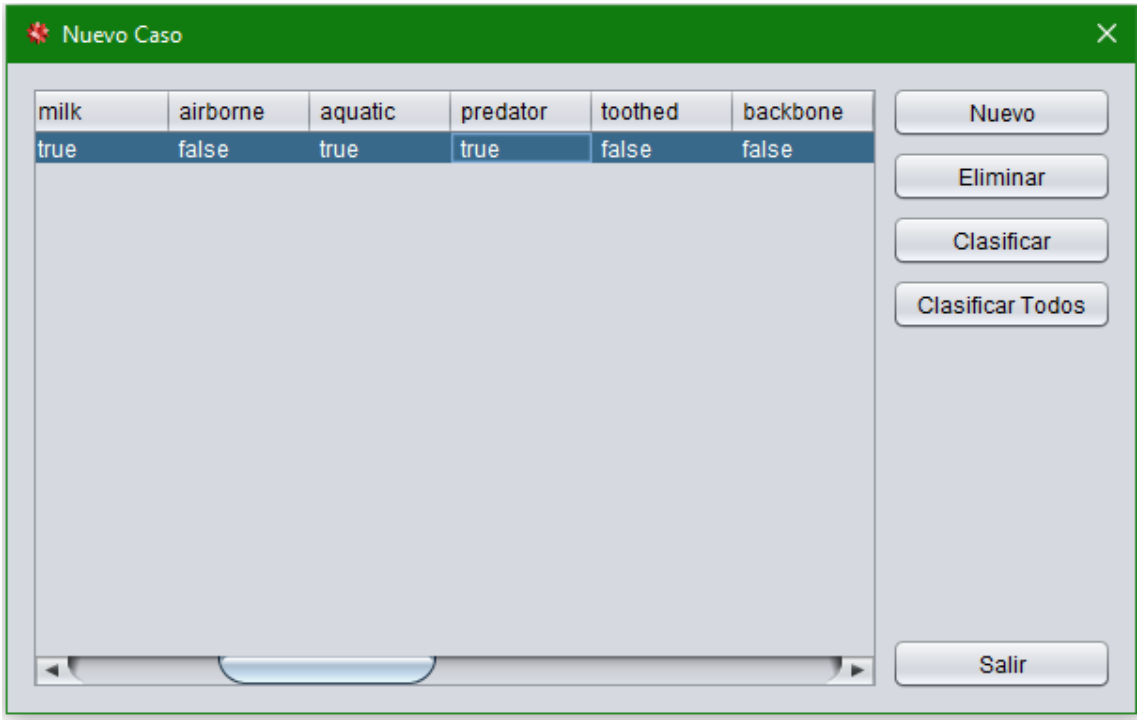


Figura 15 Clasificar nuevo caso. Fuente (Elaboración propia).

Algoritmo 2: Clasificación de nuevos objetos.

Entrada: O_n / / Objeto a clasificar

K_c / / K representa las clases y c recorre la cantidad de clases

P_c / / Conceptos de cada clase

β_0 / / Umbral de semejanza

Salida: K_i / / Clase a la que pertenece el objeto nuevo

1: Determinar a qué concepto corresponde el nuevo objeto utilizando por medio del cálculo de la ecuación 10:

$$\beta(O_n, P_c) = \frac{\sum_{i=1}^{n} \delta(x_i(O_n), x_n(P_c))}{n} \qquad (10)$$

2 $Si\ \beta(O_n, P_c) == 1, O_n \in K_C$

3: $Si\ \beta(O_n, P_c) \neq 1$

3.1: Seleccionar máximo $\beta(O_n, P_c)$

4: Si máximo $\beta(O_n, P_c) \geq \beta_0, O_n \in K_C$

5: Si máximo $\beta(O_n, P_c) < \beta_0$, se adiciona una nueva clase a la base de casos y se ejecuta el algoritmo 1.

2.4. Conclusiones parciales

Como resultado de aplicar algoritmos del Reconocimiento Lógico Combinatorio de Patrones para el agrupamiento no supervisado y tomar los conceptos como representantes de cada grupo se obtiene una estructura jerárquica de la base de casos, apoyado en los sustentos teóricos del algoritmo conceptual RGC.

La estructura propuesta considera la aplicación de métodos para la selección de rasgos, basados en la teoría de testores y se descompone en dos niveles: en el primer nivel, los conceptos que se corresponden con el agrupamiento construido y en el segundo nivel se destaca una organización de árbol jerárquico que facilita el acceso a los casos semejantes hacia el interior de cada grupo.

Capítulo 3. Experimentación y resultados

En este capítulo se realiza una descripción de las bases de casos utilizadas y se describen cada uno de los preexperimentos a realizar. Además, se muestran los resultados de aplicar el algoritmo conceptual RGC en un Sistema Basado en Casos con una estructura plana y con una estructura jerárquica conceptual de la base de casos. Se aplica el método de validación cruzada para separar los conjuntos en entrenamiento y prueba, además se aplica el test estadístico de Friedman como prueba no paramétrica.

3.1. Diseño y aplicación de preexperimentos

El propósito de la preexperimentación realizada consiste en verificar la veracidad de la hipótesis:

Si se aplica el algoritmo conceptual RGC en el diseño de Sistemas Basados en Casos utilizando una estructura jerárquica conceptual para la base de casos se contribuirá a mejorar la eficacia de los Sistemas Basados en Casos para la solución de problemas de clasificación.

Entiéndase por eficacia la capacidad del Sistema Basado en Casos propuesto para realizar clasificaciones correctas.

Para analizar los resultados, se utilizaron cinco conjuntos de datos de reconocimiento internacional disponibles en el repositorio de aprendizaje automático (*UCI Machine Learning Repository*), de los cuales se conoce la cantidad de casos, cantidad de rasgos y cantidad de clases como muestra la tabla 1.

Tabla 1 Características de las bases de casos seleccionadas.

Base de Casos	Cantidad de Casos	Cantidad de Rasgos	Cantidad de Clases
Glass	214	9	7
Heart-statlog	270	13	2
Lymphography	148	18	4
Wine	177	14	3
Zoo	101	18	7

Fuente (Elaboración propia)

Se realizan 10 iteraciones de este proceso siguiendo el principio de validación cruzada (k-*fold cross validation*) con k=10. Al obtener estos resultados se utiliza el paquete del software RStudio llamado "scmamp" (*Statistical Comparison of Multiple Algorithms in Multiple Problems*) descrito en (Calvo y Santafé Rodrigo 2016) para realizar pruebas estadísticas.

Para la definición de los preexperimentos se propone realizar la prueba no paramétrica de Friedman. Este *test* no asume distribución normal ni homogeneidad en las varianzas de los datos (Demšar 2006). En particular en esta investigación se utiliza la prueba de Iman y Davenport como una extensión de la prueba de Friedman según lo sugerido en García et al. (2010).

Una vez cargada una base de casos es necesario establecer un criterio de agrupamiento para cada rasgo, un umbral de semejanza, una función de semejanza, una función objetivo y la forma de calcular los testores típicos; se procede a la construcción de la estructura jerárquica conceptual de la base de casos como se describe en el epígrafe 2.3 del capítulo anterior. La validación se divide en dos preexperimentos:

1. Determinar si existen diferencias significativas entre la clase de los nuevos casos respecto a la variable por ciento de clasificación correcta para los conjuntos de datos seleccionados utilizando una estructura plana y una estructura jerárquica conceptual de la base de casos.
2. Determinar si existen diferencias significativas entre la clase de los nuevos casos respecto a la variable por ciento de clasificación correcta para los conjuntos de datos seleccionados en comparación con algoritmos como el LC-Conceptual, K-means, Holotipo y AIC.

Preexperimento 1: Comparación entre la estructura plana y la estructura jerárquica utilizando el algoritmo conceptual RGC.

Tabla 2 Resultados del por ciento de clasificaciones correctas entre una estructura plana y la estructura jerárquica conceptual propuesta.

Base Dato	*Estructura plana*	*RGC estructura jerárquica*
Glass	96.26	96.26
Heart-statlog	97.9	98.15
Lymphography	98.68	98.68

Wine	97.08	98.07
Zoo	99.01	99.01

Fuente (Elaboración propia)

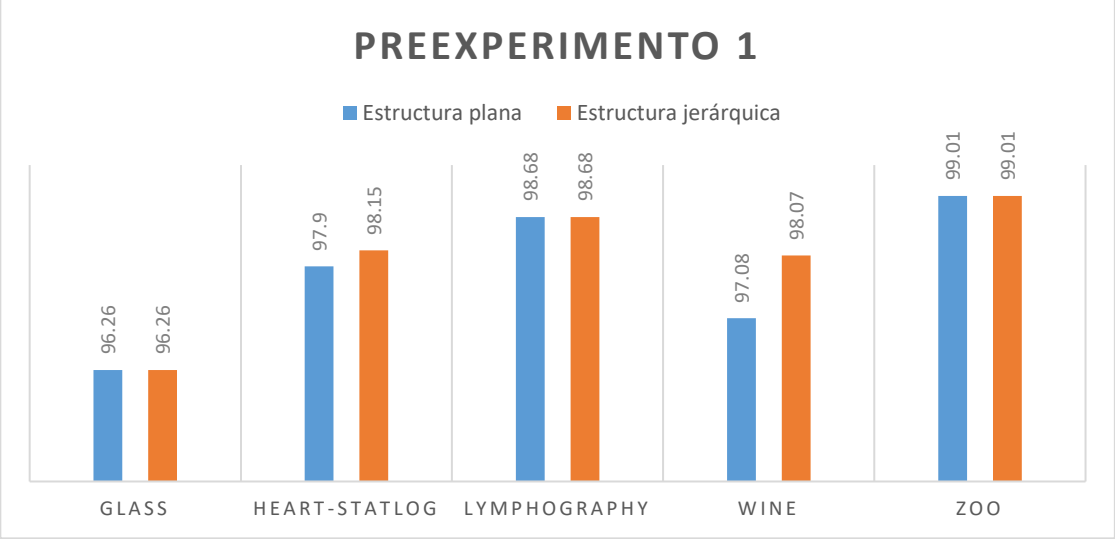


Figura 16 Comparación preexperimento 1. Fuente (Elaboración propia)

La tabla 2 y la figura 16 muestra que el comportamiento de la variable eficacia en la estructura jerárquica es similar o mejora en comparación con una estructuración plana de la base de casos.

Preexperimento 2: Comparación entre la estructura jerárquica utilizando el algoritmo conceptual RGC y otros algoritmos que optan por el uso de una estructura jerárquica.

Tabla 3 Resultados del por ciento de soluciones correctas.

Base Dato/Algoritmo	*RGC*	*LC-Conceptual*	*AIC*	*Holotipo*	*K-means*
Glass	96.26	90.65	79.56	77.45	73.89
Heart-statlog	98.15	88.14	83.7	76.6	85.96
Lymphography	98.68	89.46	83.78	87.02	86.59
Wine	98.07	97.03	94.56	95.46	94.67
Zoo	99.01	93.84	91.04	92.07	91.18

Fuente (Elaboración propia);(Reyes-González 2017)

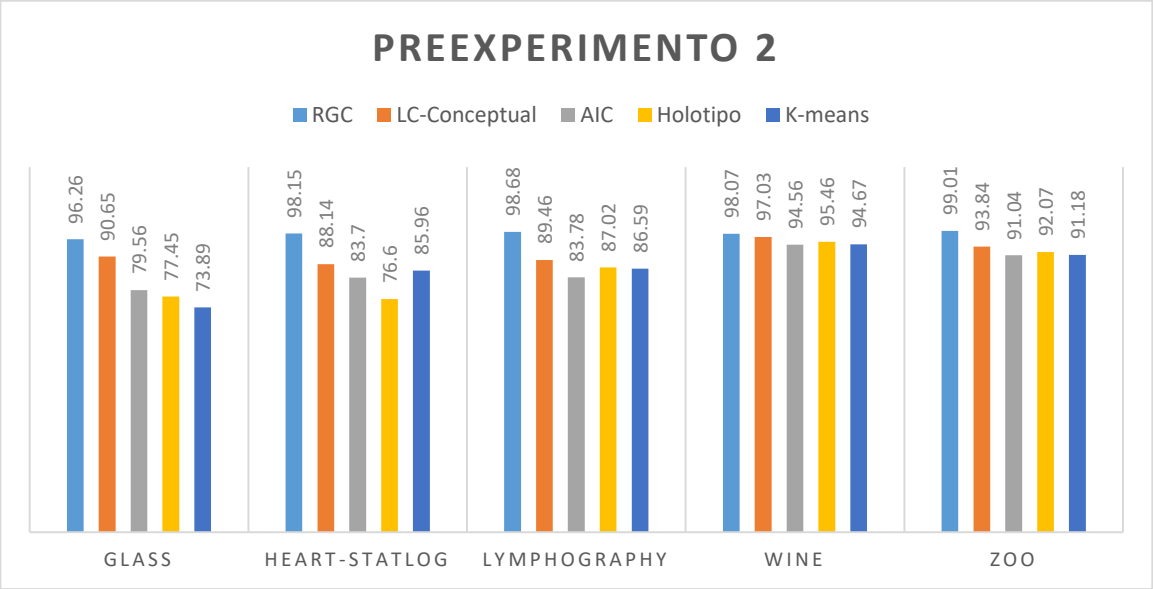


Figura 17 Comparación preexperimento 2. Fuente (Elaboración propia)

Como se evidencia en la tabla 3 y la figura 17, al aplicar el algoritmo RGC en las bases de datos seleccionadas en comparación con los resultados que se obtienen en Reyes-González (2017) mejora la eficacia en cuanto al porciento de soluciones correctas.

En la tabla 4 y la figura 18 se muestran las pruebas estadísticas aplicando el Test de Iman y Davenport con un p-value < $1.4777e^{-8}$ en un intervalo de confianza del 95%. Estos se corresponden con el porcentaje de soluciones correctas en cada una de las bases de datos seleccionadas.

Tabla 4 Ranking de Friedman.

	Ranking
1.3	RGC
1.7	Estructura Pana
3	LC-Conceptual
4.6	Holotipo
5	K-means
5.4	AIC

Fuente (Elaboración propia)

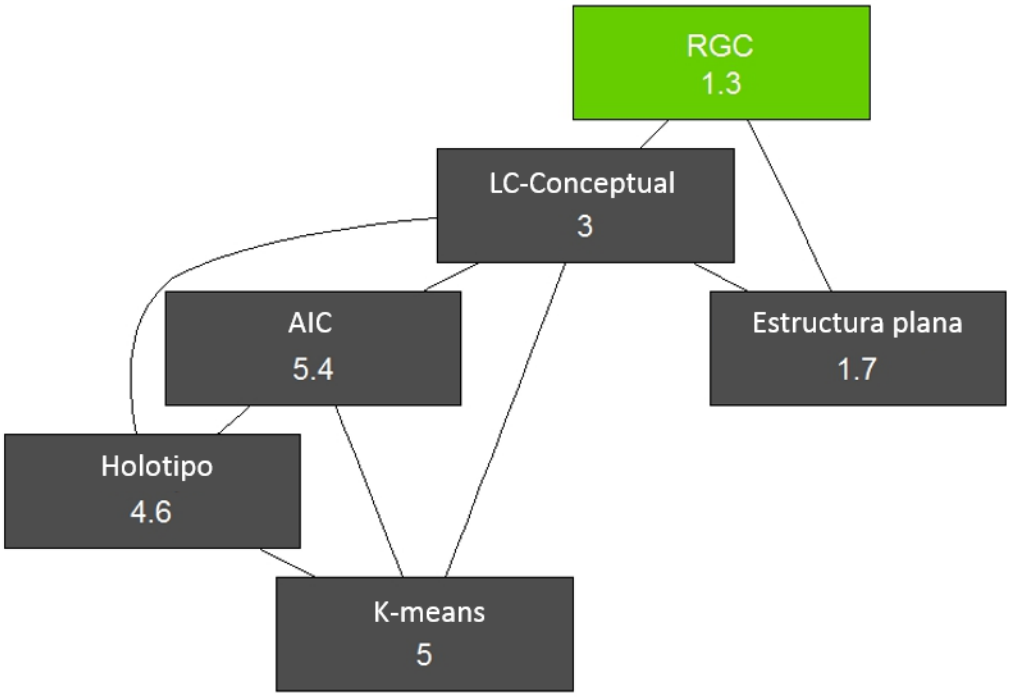


Figura 18 Ranking de Friedman. Fuente (Elaboración propia)

Asumiendo como hipótesis nula que todos los algoritmos son estadísticamente equivalentes, se aplica la prueba post hoc (ver figura 19) con la corrección de Finner (García et al. 2010) y un p-value = $1.4777e^{-8}$, como este valor está por debajo de 0.05 se debe rechazar la hipótesis nula y por tanto se puede afirmar que existen diferencias significativas entre los algoritmos.

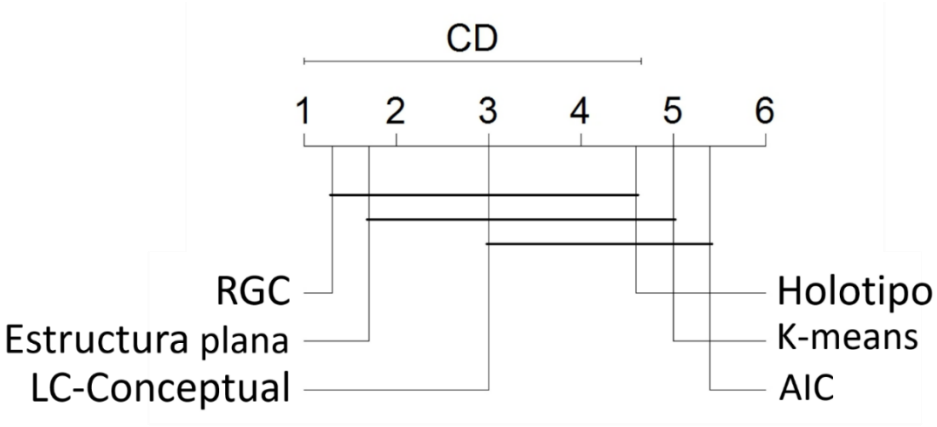


Figura 19 Diferencias significativas entre los algoritmos. Fuente (Elaboración propia)

Con los resultados obtenidos al aplicar el algoritmo RGC se mejora la eficacia en las soluciones en comparación con algoritmos como el K-means y AIC; sin embargo, con respecto a una Estructura plana, el LC-Conceptual y el Holotipo las diferencias no son significativas, aunque los resultados del RGC son superiores.

La selección de estos algoritmos para la experimentación está dada porque se considera que su funcionamiento es similar a varios de los métodos utilizados en la organización jerárquica de la base de casos que conforman conjuntos jerárquicos y luego calculan un prototipo real o artificial para comparar el caso nuevo con estos.

3.2. Caso de estudio con la base de datos Zoo

Para realizar ambos preexperimentos aplicando el algoritmo conceptual RGC se utiliza la base de datos Zoo que contiene información sobre 101 animales caracterizados por 17 variables cualitativas booleanas y una variable numérica.

Tabla 5 Descripción de rasgos de la base de datos Zoo

Rasgo	*Tipo de dato*
Animal	String
Hair	Boleano
Feathers	Boleano
Eggs	Boleano
Milk	Boleano
Airborne	Boleano
Aquatic	Boleano
Predator	Boleano
Toothed	Boleano
Backbone	Boleano
Breathes	Boleano

venomous	Boleano
Fins	Boleano
Legs	Numérico
Tail	Boleano
Domestic	Boleano
Catsize	Boleano
Type	String

Fuente (Elaboración propia)

El rasgo nombre del “Animal” no fue tenido en cuenta para la realización de los experimentos, pues el autor considera que no aporta información de importancia como variable a considerar.

Para clasificar la base de casos Zoo con el algoritmo RGC se utilizan como criterios de comparación por rasgos el de igualdad estricta para todas las variables y como función de semejanza la suma normalizada. En la fase de determinación extensional se utiliza como criterio de agrupamiento el que calcula las componentes β_o-compactas con umbral de semejanza β_o= 0.8 obteniéndose 9 agrupaciones, dos más que los grupos originales. En la fase de determinación intencional se utiliza el algoritmo FastBR a partir del peso informacional de los rasgos por frecuencia para obtener el concepto asociado a cada agrupamiento para realizar la estructura jerárquica conceptual de la base de casos (ver figura 20).

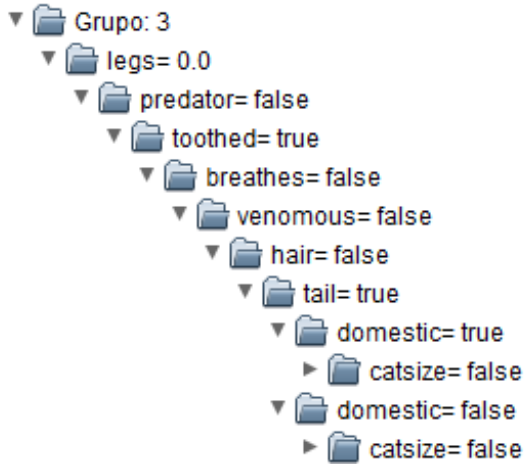


Figura 20 Estructura jerárquica del grupo 3.Fuente(Elaboración propia)

Al aplicar estas condiciones en la base de dato Zoo se compara la eficacia de la estructura plana (ver figura 19) contra la eficacia de la estructura jerárquica conceptual (ver figura 20) utilizando el algoritmo conceptual RGC.

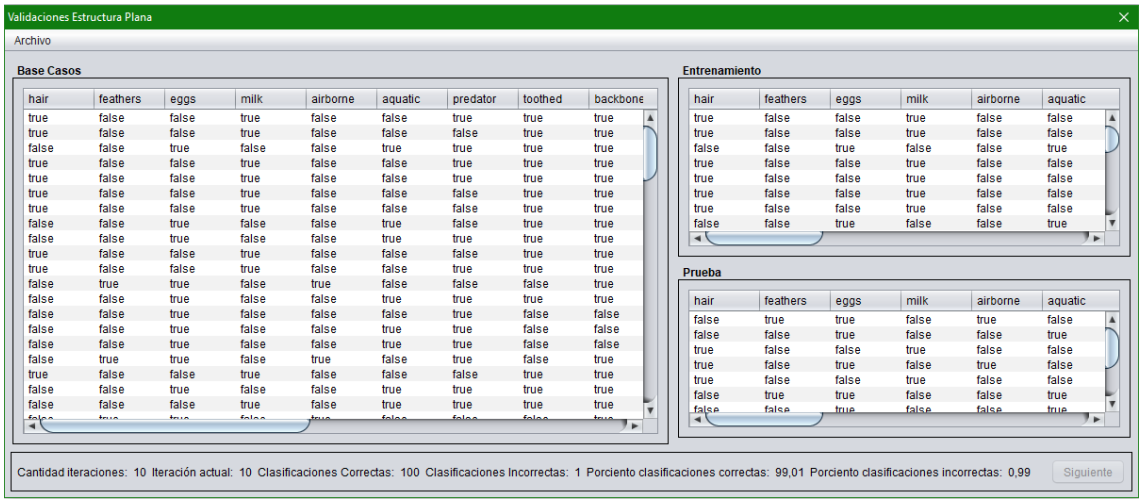


Figura 21 Validaciones estructura plana. Fuente (Elaboración propia)

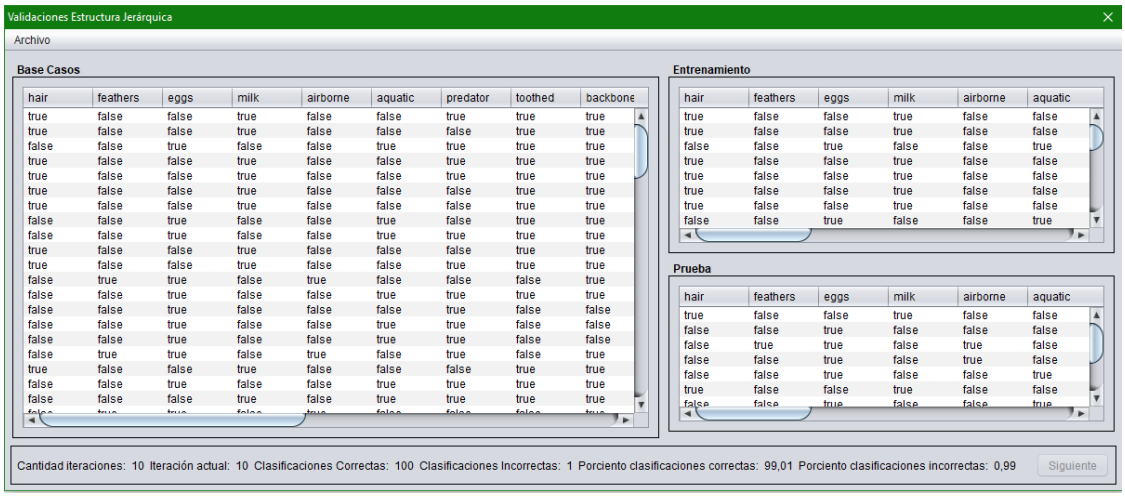


Figura 22 Validaciones estructura jerárquica. Fuente (Elaboración propia)

La figura 23 muestra los resultados luego de ejecutado el preexperimento 1, obteniéndose que no existen diferencias en cuanto a eficacia en el uso de una estructura jerárquica con respecto a una estructura plana. Sin embargo, la estructura jerárquica logra el mismo resultado con el análisis de una menor cantidad de casos.

Lo demuestra el similar comportamiento del algoritmo RGC en el acceso y recuperación de los casos semejantes, en relación a la búsqueda lineal, reportada en la literatura como la de mejores resultados.

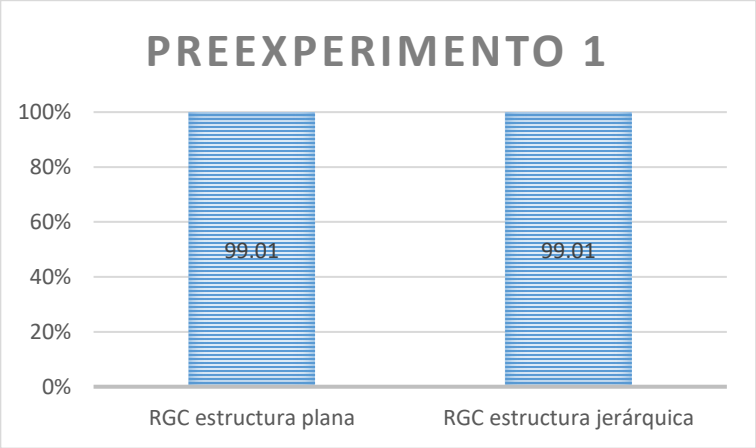


Figura 23 Resultados preexperimento 1 en el caso de estudio Zoo. Fuente (Elaboración propia)

En la figura 24 se evidencian los resultados superiores en cuanto a la eficacia, del algoritmo RGC con respecto a los algoritmos LC-Conceptual, Ideal de la Clase (AIC), Holotipo y K-means.

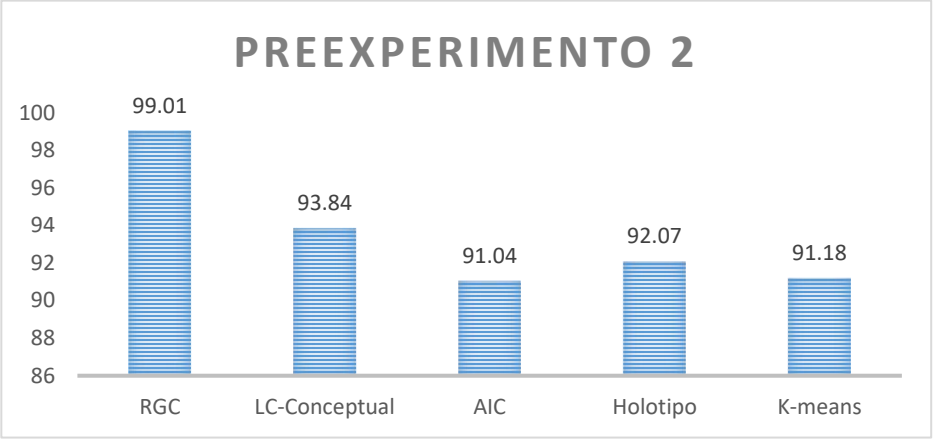


Figura 24 Resultados preexperimento 2 en el caso de estudio Zoo. Fuente (Elaboración propia)

3.3. Conclusiones parciales

Con la aplicación de los preexperimentos a las bases de datos de prueba se puede afirmar que la utilización de la estructura jerárquica conceptual de la base de casos aplicando el algoritmo conceptual RGC garantiza una mayor eficacia con respecto a otros Sistemas Basados en Casos que utilizan estructuras jerárquicas.

Se aprecian resultados significativamente superiores de la estructura jerárquica conceptual propuesta con relación a similares estructuras que conforman jerarquías de conjuntos en las bases de datos utilizadas.

Conclusiones

A partir de la sistematización de los principales referentes teóricos que sustentan la investigación, se confirma que las estructuras jerárquicas para la organización de la base de conocimiento en los Sistemas Basados en Casos, existentes en la literatura presentan limitaciones. Estas estructuras condicionan la efectividad de los métodos para el acceso y recuperación de los casos semejantes. Todo ello fundamenta la necesidad del desarrollo de una nueva estructuración jerárquica conceptual.

La estructura jerárquica conceptual propuesta, que utiliza el algoritmo RGC en el marco del Reconocimiento Lógico Combinatorio de Patrones favorece el acceso y recuperación de los casos semejantes debido a la importante propiedad que cumplen los conceptos generados para cada agrupamiento de ser caracterizantes y excluyentes.

Los métodos científicos empleados para la validación de la propuesta de solución permitieron comprobar que la solución propuesta contribuye a mejorar la eficacia respecto a otros tipos de sistemas basados en casos que emplean organizaciones jerárquicas para la base de casos.

Recomendaciones

A partir de la investigación realizada, así como de las conclusiones generales emanadas de esta, se recomienda:

1. Evaluar la posible incorporación de métodos para el tratamiento de la incertidumbre de los casos almacenados en la base de conocimiento, así como extenderlo para el caso en que existan múltiples rasgos objetivos.
2. Aplicar las medidas de la importancia informacional de los conceptos, como heurística para la conformación de la estructura jerárquica de la base de casos.

Referencias Bibliográficas

AAMODT, A. y PLAZA, E., 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. , vol. 7, no. 1, pp. 39–59.

AGGARWAL, C.C. y REDDY, C.K., 2013. *Data clustering: algorithms and applications*. S.l.: CRC press. ISBN 1-4665-5821-0.

ALBA-CABRERA, E., IBARRA-FIALLO, J. y GODOY-CALDERON, S., 2013. A theoretical and practical framework for assessing the computational behavior of typical testor-finding algorithms. *Iberoamerican Congress on Pattern Recognition*. S.l.: Springer, pp. 351–358.

BANERJEE, S. y CHOWDHURY, A.R., 2015. Case based reasoning in the detection of retinal abnormalities using decision trees. , vol. 46, pp. 402–408.

BELLO, R., 2002. Aplicaciones de la inteligencia artificial. *Ediciones de la Noche, Guadalajara, Jalisco, México*, vol. 970, no. 27, pp. 0177.

BLOBEL, B., 2013. Case-based reasoning in intelligent health decision support systems. *PHealth 2013: Proceedings of the 10th International Conference on Wearable Micro and Nano Technologies for Personalized Health*. S.l.: IOS Press, pp. 44. ISBN 1-61499-268-1.

BRANTING, L.K., 2014. Integrating generalizations with exemplar-based reasoning. *Proc. of the 11th Annual Conference of the Cognitive Science Society*. S.l.: s.n., pp. 139–146.

CALVO, B. y SANTAFÉ RODRIGO, G., 2016. scmamp: Statistical comparison of multiple algorithms in multiple problems. *The R Journal*, vol. Vol. 8/1. ISSN 2073-4859.

CAO, M., ZHANG, S., YIN, Y. y SHAO, L., 2017. Classification and the case matching algorithm of the blast furnace burden surface. *AIP Conference Proceedings*. S.l.: AIP Publishing, pp. 080009. ISBN 0-7354-1488-2.

DEMŠAR, J., 2006. Statistical comparisons of classifiers over multiple data sets. , vol. 7, no. Jan, pp. 1–30.

DÍAZ-AGUDO, B. y GONZÁLEZ-CALERO, P.A., 2001. Formal concept analysis as a support technique for CBR. *Knowledge-based systems*, vol. 14, no. 3-4, pp. 163-171.

FAN, Z.-P., LI, Y.-H., WANG, X. y LIU, Y., 2014. Hybrid similarity measure for case retrieval in CBR and its application to emergency response towards gas explosion. , vol. 41, no. 5, pp. 2526–2534.

FERNANDES, F., ALVES, D., PINTO, T., TAKIGAWA, F., FERNANDES, R., MORAIS, H., VALE, Z. y KAGAN, N., 2016. Intelligent energy management using CBR: Brazilian residential consumption scenario. *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*. S.l.: IEEE, pp. 1–8. ISBN 1-5090-4240-7.

GAIA – Group of Artificial Intelligence Applications | Universidad Complutense de Madrid, Spain. [en línea], 2011. [Consulta: 29 mayo 2018]. Disponible en: https://gaia.fdi.ucm.es/.

GARCÍA, S., FERNÁNDEZ, A., LUENGO, J. y HERRERA, F., 2010. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, vol. 180, no. 10, pp. 2044-2064. ISSN 0020-0255.

GARCÍA, S., FERNÁNDEZ, A., LUENGO, J. y HERRERA, F., 2010b. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, vol. 180, no. 10, pp. 2044–2064. ISSN 0020-0255.

GOYAL, R. y SRIVASTAVA, D.K., 2016. A Study on Cluster Analysis Technique-Hierarchical Algorithms. , vol. 2, no. 9.

GRAUPE, D., 2013. *Principles of artificial neural networks*. S.l.: World Scientific. ISBN 981-4522-74-0.

GUO, Y., HU, J. y PENG, Y., 2014. Research of new strategies for improving CBR system. , vol. 42, no. 1, pp. 1–20.

GUZMÁN-TRAMPE, J.E., 2009. *Extensión de un lenguaje formal (LCARS) para especificar problemas de clasificación y de selección de rasgos, mediante la construcción de un intérprete optimizado.* 2009. S.l.: s.n.

HAN, M. y CAO, Z., 2015. An improved case-based reasoning method and its application in endpoint prediction of basic oxygen furnace. , vol. 149, pp. 1245–1252.

HERRERO, I., URDIALES, C., PEULA, J.M. y SANDOVAL, F., 2015. A bottom-up robot architecture based on learnt behaviors driven design. *International Work-Conference on Artificial Neural Networks*. S.l.: Springer, pp. 159–170.

HUANG, M.-L., HUNG, Y.-H., LEE, W.-M., LI, R.K. y WANG, T.-H., 2012. Usage of case-based reasoning, neural network and adaptive neuro-fuzzy inference system classification techniques in breast cancer dataset classification diagnosis. , vol. 36, no. 2, pp. 407–414.

HUI, L.I., WEI-SHU, M.A., YI, X.I.N. y YAN-LI, L.U., 2016. Research of Hospital Knowledge Management Based on Case-Based Reasoning Technology. *DEStech Transactions on Social Science, Education and Human Science*, no. icss.

IGNIZIO, J.P., 1991. *Introduction to expert systems: the development and implementation of rule-based expert systems*. S.l.: McGraw-Hill. ISBN 0-07-112646-5.

KAKLAUSKAS, A., 2015. Intelligent decision support systems. *Biometric and Intelligent Decision Making Support*. S.l.: Springer, pp. 31–85.

KANG, Y.-B., KRISHNASWAMY, S. y ZASLAVSKY, A., 2014. A retrieval strategy for case-based reasoning using similarity and association knowledge. , vol. 44, no. 4, pp. 473–487.

KHAMPARIA, A. y PANDEY, B., 2017. A novel method of case representation and retrieval in CBR for e-learning. , vol. 22, no. 1, pp. 337–354.

KIM, B., RUDIN, C. y SHAH, J.A., 2014. The bayesian case model: A generative approach for case-based reasoning and prototype classification. *Advances in Neural Information Processing Systems*. S.l.: s.n., pp. 1952–1960.

KOLODNER, J., 1993. 10 - Indexing and Retrieval. En: DOI: 10.1016/B978-1-55860-237-3.50016-9, *Case-Based Reasoning* [en línea]. San Francisco (CA): Morgan Kaufmann, pp. 369–389. ISBN 978-1-55860-237-3. Disponible en: https://www.sciencedirect.com/science/article/pii/B9781558602373500169.

KOLODNER, J.L., 1992. An Introduction to Case-Based Reasoning'Artificial Intelligence Review 6, 3–34. ,

LIAS-RODRIGUEZ, A. y SANCHEZ-DIAZ, G., 2013. An algorithm for computing typical testors based on elimination of gaps and reduction of columns. , vol. 27, no. 8, pp. 1350022.

LI, H., SONG, Y., LI, X., LIU, Q. y ZHU, Y., 2015. Research of CBR retrieval method based on rough set theory. *Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on*. S.l.: IEEE, pp. 990–993. ISBN 1-4799-8353-5.

LOPEZ DE MANTARAS, R.L., MCSHERRY, D., BRIDGE, D., LEAKE, D., SMYTH, B., CRAW, S., FALTINGS, B., MAHER, M.L., T COX, M. y FORBUS, K., 2005. Retrieval, reuse, revision and retention in case-based reasoning. , vol. 20, no. 3, pp. 215–240.

MAHER, M.L. y PU, P., 2014. *Issues and Applications of Case-Based Reasoning to Design.* S.l.: Psychology Press. ISBN 978-1-317-77891-2.

MARTÍNEZ-TRINIDAD, J.F. y RUIZ-SHULCLOPER, J., 1999. LC-conceptual algorithm: characterization using typical testors by class. *Proceedings of the 7th European Congress on Intelligent Techniques & Soft Computing*. Aache, Germany: s.n.,

MARTÍNEZ-TRINIDAD, J.F. y SÁNCHEZ-DÍAZ, G., 2001. LC: a conceptual clustering algorithm. *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. S.l.: Springer, pp. 117–127.

MICHALSKI, R., 1980. Knowledge acquisition through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts. , vol. 4, pp. 219–243.

MICHALSKI, R.S., 1979. Conceptual clustering: a theoretical foundation and a method for partitioning data into conjunctive concepts. ,

MICHALSKI, R.S. y STEPP, R.E., 1981. *Concept-based clustering versus numerical taxonomy*. S.l.: Department of Computer Science, University of Illinois at Urbana-Champaign.

MÜLLER, G. y BERGMANN, R., 2014. A cluster-based approach to improve similarity-based retrieval for Process-Oriented Case-Based Reasoning. *Proceedings of the Twenty-first European Conference on Artificial Intelligence*. S.l.: IOS Press, pp. 639–644. ISBN 1-61499-418-8.

myCBR. [en línea], 2006. [Consulta: 29 mayo 2018]. Disponible en: http://www.mycbr-project.net/.

ORACLE CORPORATION, 2018. Bienvenido a NetBeans y www.netbeans.org, Portal del IDE Java de Código Abierto. [en línea]. [Consulta: 17 mayo 2018]. Disponible en: https://netbeans.org/index_es.html.

PAL, S.K., DILLON, T.S. y YEUNG, D.S., 2012. *Soft computing in case based reasoning*. S.l.: Springer Science & Business Media. ISBN 1-4471-0687-3.

PERNER, P., 2014. Mining sparse and big data by case-based reasoning. , vol. 35, pp. 19–33.

PERNER, P., 2017. Model Development and Incremental Learning Based on Case-Based Reasoning for Signal and Image Analysis. . Cham: Springer International Publishing, pp. 3–24. ISBN 978-3-319-54609-4.

PEULA, J.M., BALLESTEROS, J., URDIALES, C. y SANDOVAL, F., 2017. Biomimetic Navigation Using CBR. *International Work-Conference on Artificial Neural Networks*. S.l.: Springer, pp. 632–643.

PICO-PEÑA, R., 1995. *PROGNOSIS. Sistema Herramienta de Reconocimiento de Patrones.* 1995. S.l.: s.n.

PONS-PORRATA, A., 2004. *DESARROLLO DE ALGORITMOS PARA LA ESTRUCTURACIÓN DINÁMICA DE INFORMACIÓN Y SU APLICACIÓN A LA DETECCIÓN DE SUCESOS.* Trabajo de Tesis en opción al grado científico de Doctor en Ciencias Técnicas. S.l.: s.n.

RECIO-GARCÍA, J.A., GONZÁLEZ-CALERO, P.A. y DÍAZ-AGUDO, B., 2014. jcolibri2: A framework for building Case-based reasoning systems. , vol. 79, pp. 126–145.

REYES-GONZÁLEZ, Y., 2014. *Modelo para la adaptación de las soluciones en un Sistema Basado en Casos utilizando el agrupamiento conceptual*. S.l.: Tesis de Maestría.

REYES-GONZÁLEZ, Y., 2017. *Modelo basado en casos utilizando algoritmos conceptuales del Reconocimiento Lógico Combinatorio de Patrones*. Tesis Doctorado. La Habana: Universidad de las Ciencias Informáticas.

REYES-GONZÁLEZ, Y., ARCEO, A.C., MARTÍNEZ-SÁNCHEZ, N. y HERNÁNDEZ-DOMÍNGUEZ, A., 2016. Combinatorial logic conceptual clustering: an alternative to Decision Making. , vol. 19, no. 57, pp. 82–96.

REZVAN, M.T., HAMADANI, A.Z. y SHALBAFZADEH, A., 2013. Case-based reasoning for classification in the mixed data sets employing the compound distance methods. , vol. 26, no. 9, pp. 2001–2009.

RICHTER, M.M. y WEBER, R.O., 2013. Case-Based Reasoning: A Textbook. *Springer Science & Business Media*, ISSN 978-3-642-40167-1.

R: The R Project for Statistical Computing. [en línea], 2018. [Consulta: 2 junio 2018]. Disponible en: https://www.r-project.org/.

RUIZ-SHULCLOPER, J., 2009. Reconocimiento lógico combinatorio de patrones: teoría y aplicaciones (Tesis en opción al grado científico de Doctor en Ciencias). ,

RUIZ-SHULCLOPER, J., 2013. Acerca del surgimiento del Reconocimiento de Patrones en Cuba. , vol. 7, no. 2, pp. 169–192.

RUIZ-SHULCLOPER, J. y MARTÍNEZ-TRINIDAD, J.F., 1995. Clasificación Sin Aprendizaje y Con Aprendizaje Parcial (Enfoque Lógico Combinatorio). *Grupo de Reconocimiento de Patrones Cuba-México. Centro de Investigación y de Estudios Avanzados del IPN Dpto de Ingeniería Eléctrica, México*,

R: What is R? [en línea], 2018. [Consulta: 30 mayo 2018]. Disponible en: https://www.r-project.org/about.html.

SANCHEZ-DIAZ, G., DIAZ-SANCHEZ, G., MORA-GONZALEZ, M., PIZA-DAVILA, I., AGUIRRE-SALADO, C.A., HUERTA-CUELLAR, G., REYES-CARDENAS, O. y CARDENAS-TRISTAN, A., 2014. An evolutionary algorithm with acceleration operator to generate a subset of typical testors. , vol. 41, pp. 34–42.

SANTANA, A. y NIEVES-HERNÁNDES, C., 2018. *Presentación del Curso: El entorno estadístico R (R4ULPGC)* [en línea]. Gran Canaria: Universidad de las Palmas. [Consulta: 30 mayo 2018]. Disponible en: http://www.dma.ulpgc.es/profesores/personal/stat/cursoR4ULPGC/1-presentacion.html.

SARKHEYLI, A. y SÖFFKER, D., 2015. Case indexing in Case-Based Reasoning by applying Situation Operator Model as knowledge representation model. *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 81-86.

SCHANK, R.C. y ABELSON, R.P., 2013. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. S.l.: Psychology Press. ISBN 1-134-91966-2.

SCHANK, R.C. y RIESBECK, C.K., 2013. *Inside computer understanding: Five programs plus miniatures*. S.l.: Psychology Press. ISBN 1-135-83039-8.

SHARAF-EL-DEEN, D.A., MOAWAD, I.F. y KHALIFA, M.E., 2014. A new hybrid case-based reasoning approach for medical diagnosis systems. , vol. 38, no. 2, pp. 9.

SHOKOUHI, S.V., SKALLE, P. y AAMODT, A., 2014. An overview of case-based reasoning applications in drilling engineering. , vol. 41, no. 3, pp. 317–329.

SINGH, P., SINGH, A.P. y AHMAD, S., 2016. Case based reasoning model in the diagnosis of psychiatric disorder. *Communication and Electronics Systems (ICCES), International Conference on*. S.l.: IEEE, pp. 1–6. ISBN 1-5090-1066-1.

SUN, G., SAWARAGI, T., HORIGUCHI, Y. y NAKANISHI, H., 2014. Knowledge-Intensive Teaching Assistance System for Industrial Robots Using Case-Based Reasoning and Explanation-Based Learning. , vol. 47, no. 3, pp. 4535–4540.

SUN, J., ZHAO, Q., ANTONY, S. y CHEN, S., 2015. Personalized Recommendation Systems: An Application in Case-based Reasoning. ,

YERO-OSES, E.A., REYES-GONZÁLEZ, Y. y MARTÍNEZ-SÁNCHEZ, N., 2016. *CEPAR: Un sistema herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones.* S.l.: s.n.