



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 2**

---

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

---

*Algoritmos de escala interior para el cálculo de los  
testores típicos como extensión de la herramienta  
CEPAR.*

**AUTOR:**

- **Leonel González Ung**

**TUTORES:**

- **Ing. Ernesto Alejandro Yero Osés**
- **Lic. Lester Rodríguez Vallejo**

**La Habana, junio de 2016**

**“Año 58 de la Revolución”**



*"Hay tanto que decir, que ha de decirse en el menor número de palabras posibles: eso sí, que cada palabra lleve ala y color"*

*José Martí*

---

## DECLARACIÓN DE AUTORÍA

---

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente al 1 día del mes de julio del año 2016.

---

Leonel González Ung

Firma del Autor

---

Ing. Ernesto Alejandro Yero Oses

Firma del Tutor

---

Lic. Lester Rodríguez Vallejo

Firma del Tutor

---

## DATOS DE CONTACTO

---

**Ing. Ernesto Alejandro Yero Oses** ([yero@uci.cu](mailto:yero@uci.cu)): graduado de Ingeniería en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas en el año 2015. Pertenece al Centro de Informática Médica (CESIM). Investiga sobre el Reconocimiento Lógico Combinatorio de Patrones.

**Lic. Lester Rodríguez Vallejo** ([lesterr@uci.cu](mailto:lesterr@uci.cu)): graduado en Ciencias de la Computación en la Universidad Central de las Villas en el año 2003. Profesor Asistente del departamento de Programación y Sistemas Digitales de la Facultad 2 en la UCI. Jefe de la disciplina Técnicas de Programación de Computadoras de la UCI. Investiga sobre el Reconocimiento Lógico Combinatorio de Patrones y sobre la Teoría de los Conjuntos Aproximados.

---

## AGRADECIMIENTOS

---

*Le agradezco primero que todo, a mis padres por ser las personas más importantes de mi vida, las cuales me han apoyado durante todos estos años de carrera y vida. A mi madre por ser la persona que más quiero y de la cual me siento más que orgulloso de tenerla como madre y a la cual espero algún día poder devolverle todo lo que me ofrecido durante estos años de vida.*

*A mi padre, por ser la persona que más me ha apoyado durante estos años de carrera, el cual nunca se ha cansado de darme consejos y todo su apoyo. Al cual le estoy agradecido por saber ser un padre, exigiéndome ser una mejor persona cada día y a el cual le debo todo lo que soy.*

*A mi hermana, por siempre estar al pendiente de mí, dándome todo su apoyo y cariño durante estos años de vida.*

*A mis viejos y mejores amigos, Mario y Odelvis, los cuales considero como una familia. Son de los buenos amigos que me llevo de estos años de carrera, con los cuales compartí los mejores momentos de mi vida, gracias por su apoyo.*

*A mis amigos Aniel, Raúl, Bryan, Judith, Yartiza con los cuales compartí el mayor tiempo de la carrera, y de los cuales solo guardo buenos recuerdos.*

*A mis amigos Andy, Abraham a pesar de que los conocí en los últimos momentos de la carrera, se convirtieron en muy buenos amigos.*

*A mis compañeros de cuarto David, Ramón y Viset.*

*A mi tutor, por apoyarme durante estos últimos meses en la universidad y por guiarme en la elaboración de esta tesis.*

*A mis amigos y hermanos de la infancia, Luis y Alejandro.*

*Por último, agradecerle a un grupo de personas que siempre han estado presente, durante todos años de carrera, sin las cuales no hubiera logrado alcanzar mis metas, las quiero y gracias por todo.*

---

## DEDICATORIA

---

*La presente tesis se la dedico con amor y cariño, a las dos personas que más se han sacrificado y dado su amor de forma incondicional durante estos años de carrera, a mis padres, Mario y Lourdes. Gracias, por creer en mí en todo momento y por saber convertirme en una persona con buenos sentimientos y valores. Gracias, por darme la posibilidad de superarme como persona y como profesional.*

---

## RESUMEN

---

El Reconocimiento Lógico Combinatorio de Patrones, es uno de los enfoques de Reconocimiento de Patrones, resalta por permitir el trabajo simultáneo con variables cuantitativas y cualitativas. Los problemas de selección de rasgos pretenden identificar las características más relevantes que influyen en un fenómeno, en el Reconocimiento Lógico Combinatorio de Patrones se resuelve apoyándose en el peso informacional de los rasgos, comúnmente apoyados por la Teoría de Testores (Testores Típicos).

En la actualidad existen herramientas para apoyar el trabajo del RP, pero de estas muy pocas están desarrolladas desde el RLCP, estas a su vez no poseen varios algoritmos para el cálculo de los Testores Típicos, encontrándose estos dispersos entre las herramientas. Los investigadores pierden tiempo adaptando los datos a cada una de las herramientas que posee, ocasionando variaciones en el modelo original, esto conlleva que los resultados no puedan ser comparados con total exactitud; ocasionando dificultades en la selección del mejor algoritmo para un problema dado.

Se presenta como propuesta de solución al problema antes descrito la incorporación de algoritmos para el cálculo de todos los Testores Típicos en la herramienta CEPAR, facilitando a los investigadores de este enfoque, la selección del mejor algoritmo para calcular todos los Testores Típicos. La incorporación de los algoritmos en la herramienta garantizará su reutilización en los problemas que empleen la herramienta CEPAR para el Reconocimiento Lógico Combinatorio de Patrones.

**Palabras clave:** Reconocimiento Lógico Combinatorio de Patrones, Testores Típicos, Sistema Herramienta.

---

## ÍNDICE

---

Introducción .....	12
Capítulo 1 Fundamentación Teórica .....	17
1.1 Reconocimiento de Patrones.....	17
1.1.1 Conceptos básicos asociados al Reconocimiento de Patrones. ....	19
1.1.2 Problemas del Reconocimiento de Patrones.....	21
1.2 La selección de rasgos y objetos. ....	22
1.3 Teoría de Testores. ....	23
1.3.1 Peso informacional de los rasgos en función del concepto de testor.....	23
1.4 Algoritmos para el cálculo de todos los Testores Típicos.....	25
1.4.1 Algoritmos de Escala interior para el cálculo de Testores Típicos. ....	25
1.4.2 Justificación de la selección de los algoritmos .....	28
1.5 Sistemas Herramientas para el Reconocimiento de Patrones. ....	29
1.5.1 Sistema ACLAPIP .....	30
1.5.2 Sistema PROGNOSIS.....	31
1.5.3 Sistema SELECTOR .....	32
1.5.4 Sistema ALISA.....	33
1.5.5 Software WEKA .....	33
1.5.6 Sistema CEPAR.....	34
1.5.7 Justificación de la selección .....	35
1.6 Herramientas y tecnologías que se utilizan en la propuesta de solución.....	36
1.6.1 Entorno de desarrollo integrado (IDE). ....	36
1.6.2 Lenguaje de programación.....	37
1.6.3 Herramienta Case: Visual Paradigm: .....	37
Capítulo 2 Propuesta de solución para el cálculo de los testores típicos en problemas de selección de rasgos .....	38
2.1 Descripción e Implementación.....	38
2.1.1 Algoritmo CT. ....	38

2.1.2 Algoritmo Fast-CT.....	40
2.2 Integración de los algoritmos en la herramienta.....	43
Capítulo 3 Validación de la propuesta de solución .....	49
3.1 Proceso de validación de la solución .....	49
3.2 Validación por expertos .....	50
3.2.1 Método Delphi.....	50
3.2.2 Proceso de selección de los expertos .....	51
3.2.3 Elaboración del cuestionario .....	54
3.2.4 Análisis y procesamiento de los resultados .....	55
3.3 Validación de los algoritmos .....	57
3.4 Validación de satisfacción de usuarios potenciales .....	59
3.4.1 Aplicación de la técnica ladov .....	59
3.4.2 Análisis de los resultados.....	61
3.5 Resultados de las pruebas y validaciones .....	62
Conclusiones .....	63
Recomendaciones .....	64
Referencias.....	65
Anexos.....	73
Anexo 1 Encuesta realizada a los especialistas .....	73
Anexo 2 Encuesta realizada a estudiantes y profesores del reconocimiento de patrones. .....	74
Anexo 3 Análisis de las herramientas existentes para el RLCP.....	76

---

## ÍNDICE DE ECUACIONES

---

Ecuación 1 :Peso informacional del rasgo.....	24
Ecuación 2: Peso del rasgo dependiendo de la longitud del testor.....	24
Ecuación 3 Coeficiente de Conocimiento.....	52
Ecuación 4 Coeficiente de Argumentación.....	53
Ecuación 5 Coeficiente de competencia.....	53
Ecuación 6 Coeficiente de Kendall.....	55
Ecuación 7 Chi-Cuadrado.....	56
Ecuación 8 Función de semejanza.....	58
Ecuación 9 Umbral de semejanza.....	58
Ecuación 10 Índice de satisfacción grupal.....	60

---

## ÍNDICE DE ILUSTRACIONES

---

Figura 1 Enfoques del Reconocimiento de Patrones.....	18
Figura 2 Flujo de los algoritmos de escala interior. ....	26
Figura 3 Diagrama de clases del algoritmo CT.....	44
Figura 4 Diagrama de clases del algoritmo Fast-CT.....	45
Figura 5 Paquetes de CEPAR.....	45
Figura 6 Paquetes de CEPAR con los algoritmos integrados.....	45
Figura 7 Herramienta CEPAR .....	46
Figura 8 Menú de selección cálculo de testores típicos.....	46
Figura 9 Seleccionar algoritmo para el cálculo de los testores típicos.....	47
Figura 10 Resultados del algoritmo seleccionado. ....	47
Figura 11 Procesos de validación .....	49
Figura 12 Coeficiente de competencia. ....	53
Figura 13 Gráfica de adecuación de preguntas.....	57
Figura 14 Porcentaje de nivel de adecuado. ....	57
Figura 15 Índice de satisfacción grupal. ....	61
Figura 16 Características de los encuestados de ladov. ....	61
Figura 17 Resultados de la aplicación de la técnica de ladov y valor de índice de satisfacción grupal(ISG).....	62
Figura 18 Porcentaje de los algoritmos. ....	76

---

## ÍNDICE DE TABLAS

---

Tabla 1 Comparación entre los algoritmos.....	29
Tabla 2 Procesamiento del grado del conocimiento sobre el tema.....	52
Tabla 3 Patrón para la determinación del nivel de argumentación. ....	52
Tabla 4 Patrón para la determinación del nivel de argumentación. ....	52
Tabla 5 Resultados del procesamiento para la determinación del coeficiente de competencia de expertos. ....	54
Tabla 6 Caracterización de los expertos que participaron en la encuesta. ....	54
Tabla 7 Resultados de la encuesta. ....	55
Tabla 8 Resultados de los algoritmos sobre un mismo juego de datos. ....	58
Tabla 9 Cuadro lógico ladov. ....	60
Tabla 10 Escala de satisfacción.....	60

---

## INTRODUCCIÓN

---

Reconocer patrones es una de las primeras actividades que aprende el ser humano, y lo utiliza para la toma de decisiones sin entender realmente lo que está haciendo. Se reconocen patrones por ejemplo al clasificar a la raza humana por el color de piel (negra, blanca, amarilla, etc.), la religión (católicos, cristianos, ortodoxos, musulmanes, etc.), la posición social (baja, media, alta, etc.) y hasta las preferencias sexuales (heterosexual, bisexual, homosexual, etc.), o para cosas mucho más especializadas que requieren un modelo matemático para representar la semejanza entre los elementos en estudio.

No es sino hasta finales de los años cincuenta, que el Reconocimiento de Patrones comienza a ser visto como una disciplina científica. La publicación del libro "*Principles of Neurodynamics*" de Rosenblatt en 1965, puede considerarse como el inicio de una disciplina que tiene como bases fundamentales a la Matemática, la Computación y las Ingenierías [1].

En sus inicios el Reconocimiento de Patrones estuvo estrechamente vinculado a la identificación, reconocimiento y clasificación de imágenes, pero con el paso del tiempo, esta disciplina ha demostrado buenos resultados, al ser aplicada a una mayor cantidad de problemas. Cuando se hable de Reconocimiento de Patrones, se estará hablando de investigaciones que tienen que ver con descripciones de objetos (de naturaleza física o abstracta), las cuales deben ser clasificadas o cuyos elementos constituyentes deben ser analizados [2]. En muchas disciplinas (Medicina, Sociología, Geociencias, Criminología y otras), se presentan problemas de clasificación, de diagnóstico, de pronóstico, de determinación de factores de influencias y otros, que son claramente problemas en los que los objetos de estudio no son necesariamente imágenes ni señales [3] [4].

El Reconocimiento de Patrones ha sido estudiado y expresado mediante varios enfoques, algunos son: Reconocimiento Estadístico de Patrones [5], Reconocimiento Sintáctico de Patrones [6], Redes neuronales [7] y Reconocimiento Lógico Combinatorio de Patrones [8]. En sentido general el Reconocimiento de Patrones resuelve problemas relacionados con la clasificación de objetos y fenómenos y con la determinación de los factores que inciden en los mismos. Así, se consideran cuatro familias de problemas que se denominan respectivamente:[9][10]

1. Selección de rasgos y/o objetos.
2. Clasificación supervisada.
3. Clasificación no supervisada.
4. Clasificación parcialmente supervisada.

La selección de rasgos y/o objetos 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 el enfoque lógico combinatorio el proceso de determinación (o selección) de los rasgos más influyentes en la clasificación de un conjunto de objetos, se apoya en el cálculo del peso informacional de los rasgos según la Teoría de Testores (a partir de los Testores Típicos).

Algunos de estos problemas están relacionados con la determinación de síndromes, factores de riesgos, perfiles de usuarios, características discriminantes de conjuntos de objetos o fenómenos, y muchos más [9]. Diversos problemas se resuelven con la selección de rasgos entre los que se pueden encontrar: En la clasificación de un complejo de géneros de hongos patógenos de la caña de azúcar [11]; En el análisis y clasificación de pacientes con fisuras de paladar [12]; En el electrodiagnóstico médico [13]; En el pronóstico gasopetrolífero [14]; En la estimación de la peligrosidad sísmica de la isla de Cuba [15]. Evidenciándose su aplicación en las más disímiles ramas de la ciencia y la sociedad.

El Reconocimiento de Patrones da sus primeros pasos en Cuba a finales de los años setenta, centrando su trabajo principalmente en dos vertientes: 1) aplicaciones matemáticas (Análisis Discriminante y Lógica Matemática), en la solución de problemas de las Geociencias y las Ciencias Sociales; 2) el procesamiento digital de señales de audio, principalmente de voz, en la Universidad Central de las Villas, en el Ministerio del Interior y en el Instituto de Cibernética, Matemática y Física (ICIMAF). A principios de los años ochenta en el (ICIMAF), existía un grupo de trabajo dedicado al procesamiento de señales de voz y otro enfocado al desarrollo de las aplicaciones de modelos lógico matemáticos para la solución de problemas de selección de rasgos y clasificación de objetos, este segundo grupo de trabajo, creó una nueva rama de investigaciones en el reconocimiento de patrones, la cual es posteriormente conocida como, Reconocimiento Lógico Combinatorio de Patrones.

En la actualidad existen varias herramientas informáticas para facilitar el trabajo en la disciplina de Reconocimiento de Patrones, pero de estas, muy pocas están desarrolladas desde la perspectiva del enfoque lógico combinatorio. Estas a su vez no poseen implementados variados algoritmos (más de dos), para el cálculo de todos los testores típicos, encontrándose dispersos en distintas herramientas (o incluso como desarrollos independientes), por lo que se presentan los siguientes problemas:

- Se dificulta el diseño de un juego de datos, en un formato admisible por las distintas herramientas, cada herramienta establece sus propios mecanismos de entrada de

información entre los que se pueden encontrar: formato de archivo atributo-relación(arff), documento en formato abierto(csv), hojas de cálculo(excel) y por último ficheros de texto(txt).

- Las características de las herramientas para la resolución de los problemas, no siempre permiten el correcto modelado del problema real.
- No es posible reproducir, con total veracidad, investigaciones que realicen comparaciones (en cuanto a: tiempos de ejecución) entre diversos algoritmos.

El investigador de esta rama pierde tiempo adaptando los datos a cada una de las herramientas, o implementando cada algoritmo. Muchas de las investigaciones en las que se realizan comparaciones entre diversos algoritmos, resultan poco reproducibles, al no existir una plataforma sobre la cual se pueda realizar este tipo de experimentos.

Ajustar el modelo del especialista a cada herramienta, con el objetivo de poder emplearla en la solución de su problema, ocasiona variaciones en el modelo original, esto conlleva que los resultados no puedan ser comparados con total exactitud, al variar el modelo para ser adaptado a cada herramienta.

Considerando lo expresado anteriormente se plantea como **problema a resolver**: ¿Cómo mejorar la elección en problemas de selección de rasgos que utilicen la Teoría de Testores para la determinación del peso informacional de los rasgos, del algoritmo para el cálculo de todos los testores típicos?

Definiéndose como **objeto de estudio**: la selección de rasgos y/o objetos en el Reconocimiento Lógico Combinatorio de Patrones y como **campo de acción** algoritmos de escala interior para el cálculo de todos los testores típicos, para la determinación del peso informacional de los rasgos.

El **objetivo general**: implementar, como extensión de un sistema herramienta para el Reconocimiento Lógico Combinatorio de Patrones, el cálculo de todos los testores típicos con algoritmos de escala interior, facilitando la selección del algoritmo a emplear en un problema dado.

Para alcanzar el objetivo planteado se proponen las siguientes **tareas de investigación**:

1. Caracterizar el marco teórico-conceptual de los algoritmos de escala interior para el cálculo de todos los Testores Típicos, así como los sistemas informáticos para la solución de problemas del Reconocimiento de Patrones haciendo énfasis en aquellos que permiten su extensión con nuevos algoritmos.
2. Seleccionar los algoritmos de escala interior para el cálculo de todos los Testores Típicos más empleados en investigaciones prácticas.

3. Identificar el sistema herramienta, que garantice una mayor reutilización de estos algoritmos, para su extensión con algoritmos de escala interior para el cálculo de todos los Testores Típicos.
4. Implementar los algoritmos de escala interior para el cálculo de todos los Testores Típicos para problemas de selección de rasgos y/o objetos, en la herramienta identificada.
5. Validar la propuesta de solución a partir de la consulta a expertos, haciendo uso de método Delphi.
6. Validar los resultados de la solución implementada mediante el desarrollo de pruebas de caja negras, a partir un caso de estudio para evaluar el conjunto de testores típicos generados por estos algoritmos.
7. Validar el nivel de satisfacción en usuarios potenciales de la propuesta desarrollada mediante el método IADOV.

Se tienen como beneficios esperados con el desarrollo de la presente investigación:

- Poseer una herramienta con variados algoritmos de escala interior para el cálculo de los testores típicos, que permita realizar comparaciones entre los resultados obtenidos.
- Permitir a los investigadores la selección del algoritmo para el cálculo de todos los testores típicos que mejor se adapte a su problema.

Para realizar las tareas de investigación se emplearon los siguientes métodos científicos:

**Teóricos:**

- **Analítico - Sintético:** en el análisis y comprensión de la teoría relacionada con el Reconocimiento de Patrones y documentación relacionada con el tema de investigación, permitiendo así extraer los conceptos e ideas fundamentales.
- **Histórico - Lógico:** en el análisis a través del tiempo del estado del arte sobre sistemas que han utilizado y utilizan algoritmos para el cálculo de los testores típicos.
- **Inductivo - Deductivo:** en la aplicación de casos de pruebas a los algoritmos, llegando a conclusiones a partir de las respuestas proporcionadas por estos.
- **Modelación:** se realizó con el objetivo de dar una descripción más detallada del funcionamiento de los algoritmos de escala interior dentro de la herramienta.

## **Empíricos:**

- **Experimentación:** A través de este método se podrá interactuar directamente con el sistema una vez integrado los algoritmos con el objetivo de identificar sus fortalezas y debilidades.
- **Entrevista:** se le realizó un conjunto de entrevistas formales no estructuradas con el fin de obtener información rápida y precisa de los principales problemas, necesidades existentes.
- **Encuesta:** Se le realizaron encuestas a un grupo de estudiantes y especialistas con el objetivo de determinar el nivel de satisfacción de los clientes mediante el método IADOV.

El trabajo realizado está estructurado en 3 capítulos, a continuación, se muestra una breve descripción de cada uno de ellos:

**Capítulo 1: Fundamentación Teórica,** se mencionan los principales conceptos necesarios para una mejor comprensión de la investigación, se realiza un análisis del estado del arte de un conjunto de herramientas existentes para el reconocimiento de patrones, además se realiza una descripción de los principales algoritmos de escala interior que existen y se describe el lenguaje de programación y entorno de desarrollo utilizados para alcanzar el objetivo propuesto.

**Capítulo 2: Propuesta de Solución,** se realiza una descripción más detallada del funcionamiento de los algoritmos de escala interior a desarrollar, para el cálculo de los testores típicos y de cómo son integrados estos a la herramienta.

**Capítulo 3: Validación de la Propuesta de Solución,** se realizarán las validaciones necesarias para evaluar los resultados de los algoritmos y se aplicarán los métodos Delphi y IADOV, con el objetivo de validar la propuesta de solución y determinar el grado de satisfacción de los clientes respecto a la propuesta de solución.

En el presente capítulo se abordan los principales conceptos asociados con el Reconocimiento Lógico Combinatorio de Patrones y la Teoría de Testores. Se realiza un análisis sobre los sistemas existentes a fines con el objetivo de esta investigación. Además, se realiza una descripción de las tecnologías, herramientas y lenguajes de programación que se utilizan para alcanzar el objetivo propuesto.

### 1.1 Reconocimiento de Patrones.

El término Reconocimiento de Patrones (RP) como muchos otros de las ciencias modernas recibe diferentes interpretaciones, incluso por los propios científicos y técnicos que se ocupan de su desarrollo.

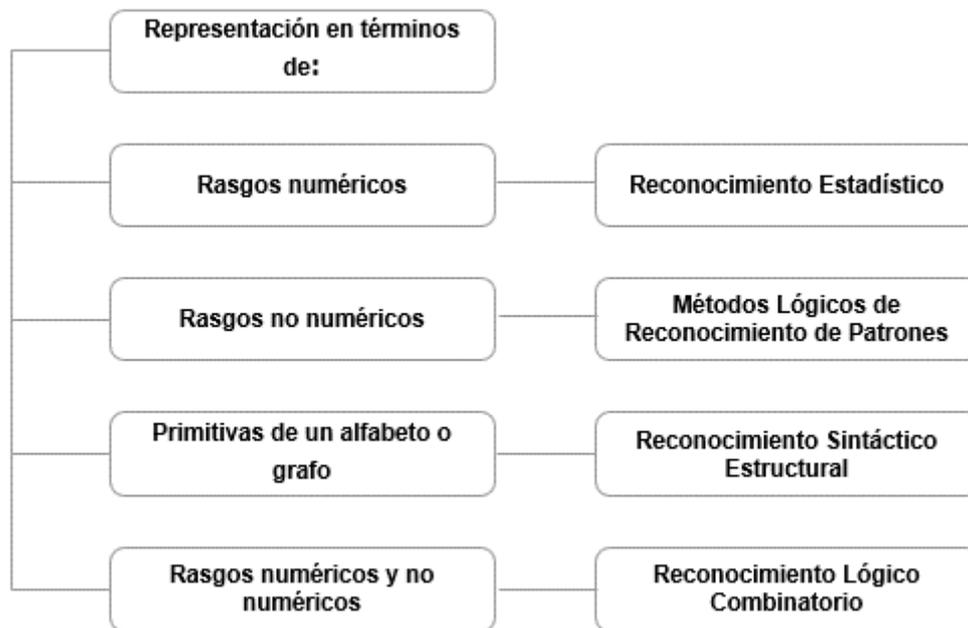
**Según José Ruíz Shulcloper:** es una ciencia con un fuerte carácter aplicado e interdisciplinario. Está relacionado con procesos (ingenieriles, físicos, matemáticos y computacionales) de datos que provienen de descripciones de objetos (fotos, hologramas, escrituras, jeroglíficos, símbolos, señales bioeléctricas, acústicas, pacientes, zonas geológicas, etc.) con el propósito de obtener (por medio de dispositivos computacionales y/o seres humanos) información que permita establecer las propiedades de ciertos subconjuntos de objetos y/o las relaciones entre ellos [9].

**Según Fukunaga:** puede ser considerado como un problema de estimación de las funciones de densidad en un alto espacio dimensional y dividir el espacio en las regiones de categorías o clases [5].

**Según Anil K. Jain:** es el estudio de cómo las máquinas pueden observando el ambiente aprender a distinguir patrones de interés de un fondo y realizar decisiones razonables sobre las categorías de los mismos [16][17].

En la presente investigación se decide adoptar el concepto de José Ruíz Shulcloper, por reflejar un concepto más amplio del Reconocimiento de Patrones y no limitarse solo a funciones básicas.

Esta variedad de criterios entorno al Reconocimiento de Patrones ha traído aparejado diversas formas (metodologías, enfoques) para resolver problemas sus distintos problemas. Atendiendo a los mecanismos utilizados para representar (modelar) estos problemas se describen cuatro enfoques como se muestra en la figura:



**Figura 1 Enfoques del Reconocimiento de Patrones.**

La presente investigación se centra en los presupuestos y teorías del Reconocimiento Lógico Combinatorio de Patrones. El enfoque lógico combinatorio facilita analizar las variables de manera diferenciada lo que permitirá que las mismas puedan ser simultáneamente de naturalezas diferentes (cuantitativas, cualitativas), e incluso permite trabajar con descripciones incompletas de objetos, como es frecuente que ocurra en muchas ramas de la Ciencia, en particular en las zonas poco formalizadas del conocimiento (medicina, geología, sociología), y como aparece en muchos problemas prácticos.

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 [18].

Es más adecuado para trabajar bajo las condiciones de descripciones mezcladas e incompletas, dado que permite captar mejor la realidad del problema y al mismo tiempo, procesar los datos mediante una modelación matemática en un marco metodológico, en el que la realidad no es desvirtuada. Esto no implica la imposibilidad de ser empleado, en casos en que las descripciones de los objetos sean exclusivamente numéricas o no numéricas, a diferencia de otros enfoques [8].

### 1.1.1 Conceptos básicos asociados al Reconocimiento de Patrones.

Para un mejor entendimiento de todos los aspectos que se abordan sobre el Reconocimiento Lógico Combinatorio de Patrones, se hace necesario definir una serie de conceptos básicos, los mismos han sido extraídos de [9]:

**Objeto** es un concepto con el cual se representa a los elementos sujetos a estudios, (un paciente de una enfermedad, una zona geográfica, un dispositivo eléctrico, un conjunto de personas, una bioseñal, una fotografía).

**Patrón** es sinónimo de objeto, solo que en algunas ocasiones es conveniente establecer diferencias entre los objetos, acerca de los cuales se conocen ciertas propiedades, por ejemplo, (la pertenencia a una clase, tipo, etc.) y patrón, de los cuales no se sabe nada acerca de dichas propiedades.

**Universo de objetos** el conjunto de todos los posibles objetos admisibles para los propósitos de estudio, en cada caso en particular.

**Clase** es un conjunto de objetos, no solo exclusivamente en el concepto matemático vinculado a la partición de un conjunto (clase de equivalencia), sino también se denomina clases, a aquellos conjuntos que formen un cubrimiento, es decir, se puede tener intersección no vacía.

**Rasgo** (variable, atributo, característica, propiedad, primitivas de un alfabeto), es un factor a tener en cuenta en el estudio de los objetos dados. Son los rasgos, la vía real para poder trabajar con los objetos, es decir, se tiene que trabajar con descripciones de los objetos en términos de un conjunto de rasgos.

**Dominio de definición** es el conjunto de valores admisibles que tiene asociado un rasgo.

A la estructura para almacenar los datos del problema (objetos, rasgos, dominios, clases) se le denominará **Matriz de Entrenamiento (ME)**, se caracteriza por tener  $m$  filas (número de objetos) y  $n$  columnas (número de rasgos), donde cada fila  $m_i$  contienen una descripción de un objeto en términos de  $n$  rasgos  $\mathfrak{R} = \{X_1, \dots, X_n\}$ , estando distribuidos en  $r$  clases  $\{C_1, \dots, C_r\}$ . Cada rasgo  $X_i$  toma valores en un conjunto  $D_i, i = 1, \dots, n$ . [19] [20].

Los mecanismos empleados para determinar qué tan semejantes son los valores de un mismo rasgo en distintos objetos son los **criterios de comparación**. Un criterio de comparación entre valores de una variable es una función  $C_i: M_i \times M_i \rightarrow L_i$  que retorna la semejanza o parecido que guardan dos valores de un mismo rasgo, siendo  $L_i$  un conjunto totalmente ordenado. Un criterio de comparación ofrece una evaluación del grado de semejanza entre dos valores cualesquiera de una variable.

La analogía o similitud es el parecido que tienen entre sí dos objetos, en dependencia de los rasgos que los describen, en el RLCP el método empleado para determinar cuan semejantes son dos objetos se denomina **función de semejanza entre objetos**. Una función de semejanza  $\Gamma$  se define como una función que se aplica a un par de objetos  $(O_i, O_j)$  obteniendo como resultado un valor real que indica qué tan semejantes son dichos objetos. Formalmente se puede definir como  $\Gamma: \mathbb{O} \times \mathbb{O} \rightarrow \mathbb{R}^+$ . Existen dos tipos de funciones de semejanza: la similitud y la disimilitud. Si es una similitud cumple que mientras más grande sea el valor de  $\Gamma(O_i, O_j)$ , más semejantes son los objetos  $O_i$  y  $O_j$ . En cambio, si es una disimilitud mientras más pequeña sea la evaluación de  $\Gamma(O_i, O_j)$ , más semejantes son los objetos [21].

Existen otros conceptos asociados al Reconocimiento de Patrones y la Teoría de Testores que se hacen preciso, dos estructuras que hacen más fácil el trabajo directo con la Matriz de Entrenamiento.

**Matriz de Diferencia (MD):** Es una matriz que se obtiene a partir de una ME y donde cada fila corresponde a la comparación rasgo a rasgo de todos los objetos de ME que se encuentran en clases diferentes, teniendo en cuenta el criterio de comparación  $\psi_i: D_i \times D_i \rightarrow \{0, 1\}$  donde se le asocia a cada  $X_i$  (0=similar, 1=disimilar), aplicando este criterio para todo posible par de objetos que pertenecen a diferentes clases de ME se crea una Matriz Booleana de Diferencia (MD) [22].

**Fila Básica:** Sean  $i_p, i_t$  filas de MD. Se dice que  $i_p$  es subfila de  $i_t$  si y sólo si:

1.  $\forall j \quad (a_{i_t j} = 0 \Rightarrow a_{i_p j} = 0)$  En todas las columnas donde  $i_p$  tiene un cero  $i_t$  también.
2.  $\exists j_0 \quad (a_{i_t j_0} = 1 \text{ y } a_{i_p j_0} = 0)$  Existe al menos una columna en la que  $i_t$  tiene un 1 y  $i_p$  no lo tiene).

También se dice que  $i_t$  es superfila de  $i_p$ . Entonces se tiene que  $i_t$  es superfila de  $i_p$ , entonces  $i_t$  es básica si y sólo si no es superfila de fila alguna de  $i_p$  en MD.

**Matriz Básica (MB):** Es una matriz compuesta por todas las filas básicas de la MD, sin repetición [23].

Los conceptos antes expuestos, constituyen la base para comprender los problemas fundamentales de los que se ocupa el Reconocimiento de Patrones.

### **1.1.2 Problemas del Reconocimiento de Patrones.**

Por problemas de Reconocimiento de Patrones de acuerdo con la bibliografía consultada [24] [25] [9] se entienden todos aquellos relacionados con la clasificación de objetos y fenómenos y con la determinación de los factores que inciden en los mismos. Así, se consideran cuatro familias de problemas que se denominan respectivamente:

1. Selección de rasgos (o variables) y objetos.
2. Clasificación supervisada (o con aprendizaje).
3. Clasificación no supervisada (o sin aprendizaje).
4. Clasificación parcialmente supervisada (o con aprendizaje parcial).

#### ***Clasificación supervisada.***

Un problema de clasificación supervisada consiste, en asumir que el universo  $U$  está estructurado en un número finito  $r$  subconjuntos propios, de cada uno de los cuales se tiene una muestra de descripciones de sus objetos. El problema consiste en encontrar las relaciones de pertenencia de un nuevo objeto de  $U$  (fuera de la muestra dada) con las  $r$  clases. Esta relación, admite la multclasificación (clases solapadas) e incluso pertenencia gradual a las clases [26].

#### ***Clasificación no supervisada.***

En este tipo de problemas no se conoce cómo se agrupan los objetos, siendo el problema principal encontrar las relaciones entre los objetos de un universo en términos de sus variables. Estas relaciones se establecen sobre la base del concepto de similaridad, teniendo como propósito agrupar los objetos según sus semejanzas. Sobre este aspecto es posible encontrar dos situaciones diferentes [27]:

- en el que se conoce el número de clases (Clasificación no supervisada restringida).
- en la que no se conoce el número de clases (Clasificación no supervisada libre).

#### ***Clasificación parcialmente supervisada.***

Este tipo de problema consiste en una combinación de los problemas de clasificación anteriormente descritos. En el universo de objetos dado, se conoce la existencia de ciertas clases e incluso se tienen muestras de algunas de ellas, pero no de todas. El problema principal es clasificar nuevos objetos en estas circunstancias, en las que no se tienen muestras de todas las clases y en las que incluso pudieran existir clases que se desconocen.

## **1.2 La selección de rasgos y objetos.**

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, para dar solución a esto se hacen un conjunto de preguntas como, por ejemplo, ¿Cuáles de ellos son “mejores” para ciertos propósitos?, ¿Cuál es la importancia informativa de cada uno de ellos y de algunos de sus subconjuntos?, ¿Cuáles rasgos tomar en cuenta?, ¿Cuál es la forma en que deben compararse los valores de dichos rasgos? Este grupo de interrogantes permitirán diferenciar entre dos problemas diferentes pero muy vinculados: la selección de rasgos para la clasificación y la selección de rasgos para la descripción [8].

En el primer grupo de problemas, lo que se representa es la determinación del mejor subconjunto de rasgos para la clasificación de nuevos objetos (no clasificados). Esto conlleva a la reducción del conjunto de todos los posibles rasgos (reducción de la dimensionalidad) sobre la base de las diferencias que estos rasgos presentan en cuanto, a clasificar a los nuevos objetos y otros problemas de optimización adicionales del subconjunto de rasgos a emplear, que también se deben tener en cuenta en muchos problemas de la realidad, como la eficiencia computacional entre otros [8].

Los problemas de selección de rasgos para la descripción son muy frecuentes en las ciencias poco formalizadas. Este problema conlleva a la determinación de un subconjunto de rasgos que de una mejor manera caracteriza a los objetos de cada una de las clases. Es claro que estos problemas asumen la existencia previa de las clases o en su defecto, la repercusión que estos rasgos tuvieron en la formación de las mismas. Problemas como la determinación de síndromes, factores de riesgos, perfiles de usuarios, características discriminantes de conjuntos de objetos o fenómenos, y muchos más, son instancias de este tipo de problemas en el mundo real [8].

### ***Peso Informativa de los Rasgos***

El peso informativo de un rasgo (o variable), es una medida de su importancia para la clasificación. En su determinación intervienen los valores de cada variable, así como los pesos informativos de estos [28].

El cálculo del peso informativo de los rasgos se realiza a partir de la Teoría de Testores, con el fin de seleccionar los rasgos que mejor describen a los objetos o los que mejor los clasifican.

### 1.3 Teoría de Testores.

La Teoría de Testores se formuló como una de las direcciones científicas independientes de la Cibernética Matemática a mediados de los años sesenta, en la ex Unión de Repúblicas Socialistas Soviéticas. Las primeras investigaciones en esta línea estaban vinculadas a la utilización de métodos lógicos matemáticos para la localización de desperfectos en los circuitos eléctricos y fueron llevadas a cabo por los especialistas soviéticos S. V. Yablonski y I. A. Cheguis. En 1965 con los trabajos de Zhuravliov, se abre una línea de aplicación de la Teoría de Testores a los problemas de Reconocimiento de Patrones [24].

**Testor:** Sea un conjunto de rasgos  $\tau = \{i_1, \dots, i_s\} \subseteq R$ , es testor de una matriz (tabla)  $T$  si al eliminar todas las columnas de  $T$  excepto las de  $\tau$ , no aparecen nuevas subdescripciones semejantes en clases diferentes [25].

**Testor Típico (TT):** Un testor se llama irreducible (típico) si al eliminar alguna de sus columnas deja de ser testor [25].

#### 1.3.1 Peso informacional de los rasgos en función del concepto de testor.

El enfoque basado en la Teoría de Testores se basa sobre la idea de que un testor típico, según Zhuravliov, es un conjunto de columnas (rasgos) de una matriz de entrenamiento para las cuales se cumple que no existen filas iguales en clases diferentes, pero además si eliminamos una cualquiera de estas columnas las restantes perderán esta propiedad [8].

La idea básica consiste en lo siguiente: dada la definición de testor y de testor típico, se tiene que cada objeto se pone en correspondencia con un s-uplo que tiene la propiedad de ser diferente en al menos alguna coordenada con respecto a cualquier otro objeto que pertenezca a una clase diferente. Al pasar de un testor a un testor típico (eliminando rasgos) se llega a una combinación irreducible, donde cada rasgo resulta imprescindible para mantener las diferencias entre las clases. Esta es la propiedad que distingue a los testores típicos. Es natural suponer pues, que, si un rasgo aparece en muchas combinaciones irreducibles, resulta más difícil prescindir de él para describir de manera diferenciante las clases, o sea que él es más diferenciante. Sobre la base de esta idea, Zhuravliov formula su definición de peso informacional de un rasgo como la frecuencia relativa de aparición de ese rasgo en la familia de todos los testores típicos [8][25].

Sea  $\tau$  el número de testores típicos que tiene una cierta matriz de entrenamiento de un problema de reconocimiento de patrones y sea  $\tau(i)$  el número de aquellos testores típicos en los que aparece la columna correspondiente al rasgo  $x_i$ . Se dice que el peso informacional (relevancia) de este rasgo viene dado por la magnitud: [25] [8]

$$P(x_i) = \frac{\tau(i)}{\tau} \text{ para } i = 1, \dots, n \text{ } x_i \in \mathbb{R}$$

### Ecuación 1: Peso informacional del rasgo

Utilizando esta definición se han resuelto una serie de problemas interesantes, sin embargo, la expresión antes propuesta por Zhuravliov para la evaluación del peso informacional de los rasgos, si bien resulta intuitiva, al considerar más importante un rasgo mientras más veces aparece en la familia de todos los testores típicos, no considera las longitudes de los testores que contienen al rasgo en cuestión, de modo que, por ejemplo, si la familia de testores típicos de un problema fuera: [25]

$$\psi^* = \{\{x_5\}, \{x_3, x_1\}, \{x_2, x_3, x_4\}\}$$

Se tiene que:

$$P(x_1) = P(x_2) = P(x_4) = P(x_5) = 0.333 \text{ y } P(x_3) = 0.666$$

Pero resulta natural pensar que el rasgo  $x_5$  es más relevante que  $x_3, x_4$  y  $x_1$  pues aunque sólo aparece en un testor típico no requiere de ningún otro rasgo para lograr diferenciar los objetos que están en clases diferentes,  $x_1$  requiere de otro rasgo y por su parte  $x_3$  y  $x_4$  requieren de otros dos; luego si se denota por  $p(x)$  la relevancia del rasgo  $x$ , podría esperarse que  $p(x_5) > p(x_1) > p(x_2) = p(x_4)$  [25]

Este análisis conduce a concluir que considerar la frecuencia de aparición de un rasgo en la familia de los testores típicos como medida de su importancia, aunque resulta bastante natural a partir de la interpretación de un testor típico como una combinación irreducible de rasgos diferenciadores, en algunos casos esto puede llevar a interpretaciones erróneas [25].

Se asociará a cada rasgo una magnitud, que depende de estas longitudes y que se denotará por: [25]

$$L(x_i) = \frac{\sum_{t \in \psi^*(x)} \frac{1}{|t|}}{|\psi^*(x)|}$$

### Ecuación 2: Peso del rasgo dependiendo de la longitud del testor.

siendo  $\psi^*(x)$  la familia de todos los testores típicos que contienen al rasgo  $x$ .

Para el ejemplo anterior se tiene que

$$L(x_5) = 1, L(x_3) = 0,418, L(x_2) = L(x_4) = 0,333 \text{ y } L(x_1) = 0,5.$$

Estos valores no se corresponden totalmente con lo que se señalaba como deseado con respecto a la relevancia, pues, aunque  $x_5$  aparece con máximo valor,  $x_1$  tiene asociado un valor mayor que el de  $x_2$  y debía esperarse lo contrario.

Entonces como ya se tiene que  $P(x)$  toma en cuenta sólo la frecuencia de aparición del rasgo en la familia de testores típicos;  $L(x)$  por su parte considera únicamente la longitud de los testores típicos en los que aparece el rasgo.

Se define una nueva medida de la relevancia del rasgo en función de estas dos magnitudes dada por la magnitud:

$$p(x) = \theta(P(x), L(x))$$

Siendo  $P(x)$  como en (ecuación 1);  $L(x)$  como en (ecuación 2) y  $\theta$  una función de estas dos magnitudes.

Considerándose la siguiente expresión como ejemplo [25]:

$$p(x) = \alpha P(x) + \beta L(x) \text{ con } \alpha, \beta > 0 \text{ y } \alpha + \beta = 1$$

siendo  $\alpha$  y  $\beta$  dos parámetros que ponderan la participación o influencia de  $P(x)$  y de  $L(x)$  respectivamente en  $p(x)$ . Esto es la importancia que se concede para la relevancia de la frecuencia de aparición y a la longitud de los testores.

Si se le concede a ambos la misma importancia se considerará  $\alpha = \beta = 0,5$ . Para estos valores en el ejemplo anterior se tiene:

$$p(x_5) = 0.666, p(x_3) = 0,542, p(x_2) = p(x_4) = 0.333 \text{ y } p(x_1) = 0.416$$

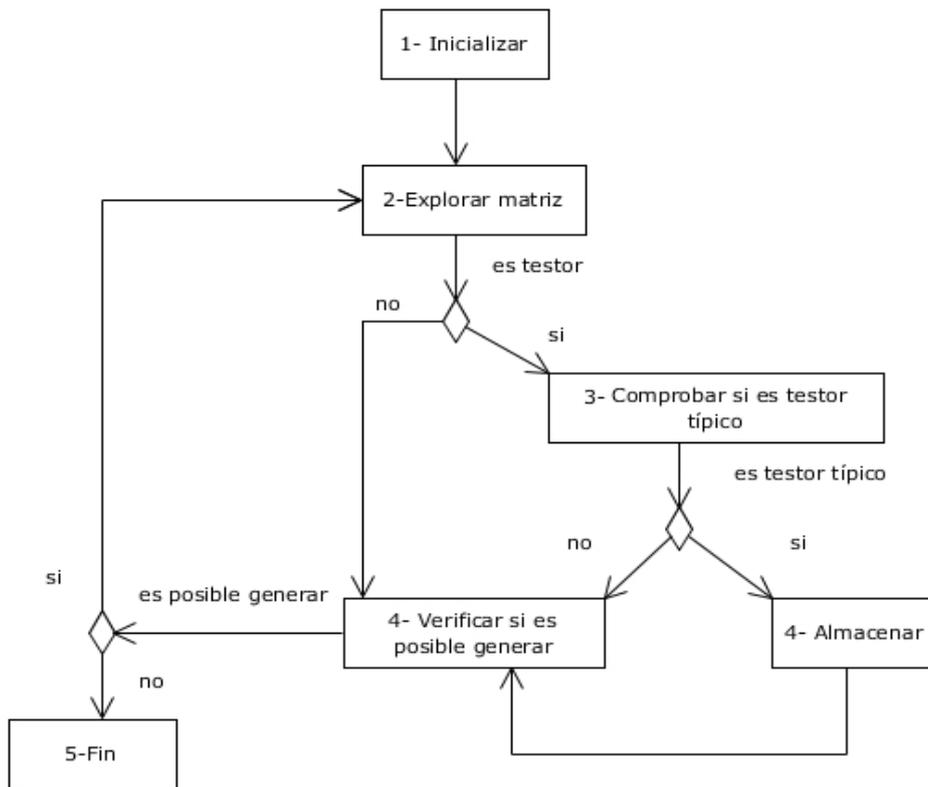
#### **1.4 Algoritmos para el cálculo de todos los Testores Típicos.**

Para el cálculo de los testores típicos se han desarrollado varios algoritmos que por su estrategia de cómputo pueden clasificarse en algoritmos de escala exterior y de escala interior.

En la presente investigación se trabajará con los algoritmos de escala interior, ya que existe otra investigación a la par, que trabaja con los algoritmos de escala exterior.

##### **1.4.1 Algoritmos de Escala interior para el cálculo de Testores Típicos.**

Los algoritmos de escala interior, construyen el conjunto de TT atendiendo a la estructura interna de la MB, teniendo en cuenta la distribución de unos y ceros en dicha matriz.



**Figura 2 Flujo de los algoritmos de escala interior.**

A continuación, se describen genéricamente el funcionamiento de los algoritmos de escala interior:

- Paso 1: Se inicializan los parámetros del algoritmo, y se ordena la MB según el algoritmo.
- Paso 2: Comprueba si es posible generar un nuevo conjunto de rasgos (a comprobar si es Testor), de no ser posible ir al *paso 7*.
- Paso 3: Se explora las posiciones unitarias de la MB, según el algoritmo hasta generar un posible Testor.
- Paso 4: Comprobar si es Testor, de no ser ir al *paso 2*.
- Paso 5: Comprobar si es Testor Típico, de no ser ir al *paso 2*.
- Paso 6: Adicionar a la lista de Testores Típicos, ir al *paso 2*.
- Paso 7: Fin del algoritmo.

**Algoritmo CT.**

El algoritmo CT [29] es de escala interior y trabaja basado en los conceptos de conjunto de filas independientes y conjunto completo. Inicialmente se encuentran los testores y luego, se prueba si son típicos o no. Por definición todo conjunto completo es un testor. El método del algoritmo CT, consiste en explorar las posiciones unitarias de la MB encontrando los conjuntos completos y comprobando, entonces, si son TT.

Para mejorar el funcionamiento del algoritmo, se le aplicó un ordenamiento previo a la MB [30]. Este cambio, en esencia, consiste en colocar como primera fila de la MB aquella que tenga la menor cantidad de valores unitarios entre todas las filas de MB y si existe más de una fila que cumpla esta propiedad, escoger aquella que tenga la mayor entropía global; o sea, aquella en la que la suma de los unos presentes en las columnas donde esta fila tiene valores unitarios sea mayor.

Éste como los demás algoritmos de escala interior, presenta la deficiencia de que repiten los TT encontrados. Este problema es inherente a su estrategia. Para que un conjunto  $\tau$  sea un TT debe existir para cada rasgo de  $\tau$  al menos una fila que tenga un 1 en esa columna y cero en todas las restantes de  $\tau$ , pero esta fila no tiene porqué ser única; de hecho, es más probable que no lo sea. Esto se traduce para el CT en que un mismo TT se origina a partir de más de un par conjunto completo-conjunto de filas independientes [29].

### ***Algoritmo FastCT.***

Es una mejora del algoritmo CT, experimentalmente se verificó que se puede realizar una mejora, comprobando todas las columnas de la matriz básica, con el fin de determinar columnas iguales, o sea que posean valores unitarios iguales en las mismas filas, en caso de existir, se eliminarán la presencia de ellas, hasta solo dejar una representación de todas. Luego se procede a ejecutar el algoritmo CT con la matriz resultante (reducida), esto permitirá reducir considerablemente los recorridos sobre la matriz. Los testores típicos que se formen de esas columnas, se llamarán pseudo\_testores, una vez que se hallan generados todos los pseudo\_testores se convertirán en testores típicos, haciendo que cada una de las columnas que fueron eliminadas sean representadas en los nuevos testores típicos, esto garantiza que el algoritmo genere la misma cantidad de testores típicos. Estas operaciones harán posible realizar una menor cantidad de recorridos sobre la matriz, al no ser necesario explorar todas las posiciones de la MB.

### ***Algoritmo CC.***

El algoritmo CC [31] es de escala interior y está basado en los conceptos de Conjunto Compatible y Conjunto Regular. Dos elementos unitarios de MB  $a_{i_1j_1}, a_{i_2j_2}$  son compatibles si  $a_{i_1j_2} = a_{i_2j_1} = 0$ . El conjunto  $D = \{a_{i_1j_1}, \dots, a_{i_sj_s}\}$  de valores unitarios de MB, es compatible si  $s = 1$  ó sus elementos son compatibles dos a dos. Un conjunto regular es un conjunto compatible, en el cual las columnas de MB asociadas a él constituyen un testor.

EL algoritmo CC tiene como peculiaridad, a diferencia del CT, que su objetivo es garantizar la tipicidad. Esto está implícito en la definición de conjunto compatible, luego tiene que comprobar si se cumple la propiedad de testor.

El algoritmo CC presenta la misma limitación señalada anteriormente al algoritmo CT, pues en este caso, un mismo TT se origina a partir de más de un conjunto compatible. Además, el algoritmo tiene otra deficiencia: para obtener un TT es necesario explorar un gran número de combinaciones de las posiciones unitarias de la MB, es decir, de conjuntos compatibles [32].

### ***Algoritmo YYC***

El algoritmo YYC [33] es un algoritmo de escala interior, que construye el conjunto de testores típicos de una MB, a partir de conjuntos que sean testores típicos para la primera fila de MB (caso booleano), incrementando este conjunto, considerando en cada paso una nueva fila de la MB.

El algoritmo tiene el siguiente funcionamiento: se comprueba la condición de testor de cada conjunto en toda la fila considerada, donde algunos de ellos pueden no cumplir la propiedad de testor, y se dirá entonces que estos conjuntos fallan; entonces se crean nuevos conjuntos que se forman por la unión de los conjuntos que fallan y un rasgo correspondiente a algún elemento no nulo de la nueva fila. Para garantizar que estos conjuntos así formados sean típicos, basta considerar solamente aquellos elementos de la nueva fila que sean compatibles con los elementos del conjunto compatible que soporta al testor.

El algoritmo al incorporar cada fila, divide la lista de testores típicos en dos: una lista de testores típicos “buenos”, que cumplen la condición de testor para esta nueva fila, y la lista de testores típicos “malos” unido con el rasgo correspondiente a cada valor no nulo de esa fila tomando ese valor.

En la presente investigación se decide implementar el algoritmo CT, por ser uno de los algoritmos básicos y principales de escala interior y el algoritmo Fast-CT por ser una mejora de los algoritmos de escala interior en este caso del CT, además después de haber realizado las encuestas(ver [anexo 2](#)), se determinó que el algoritmo CT no puede estar ausente entre los algoritmos a desarrollar para la presente investigación puesto que lo consideran uno de los más importantes para el cálculo de los testores y para la investigación, siendo identificado entre los algoritmos señalados como el más conocido en su escala.

#### **1.4.2 Justificación de la selección de los algoritmos**

Los algoritmos CT y CC resultan ser, como se observó en la descripción anterior, los algoritmos básicos a partir de los cuales se han desarrollado nuevas mejoras. Cabe resaltar según un estudio realizado en [34], los distintos tiempos de ejecución de ambos algoritmos sobre una misma MB.

Orden de MB	CC	CT
22x15	14832 mseg	571 mseg
24x16	39697 mseg	2824 mseg
29x17	3 min	4457 mseg
54x20	1 h	4 min
56x25		2h

**Tabla 1 Comparación entre los algoritmos**

En la presente investigación se decide implementar tomando en cuenta el tiempo de ejecución y el hecho de que es uno de los algoritmos básicos el CT, y de igual forma se decide incluir su versión Fast-CT que contempla una mejora en el procesamiento de la MB.

Se tiene además como resultado de la encuesta (ver [anexo 2](#)) aplicada que los algoritmos de escala interior más conocidos son el CT y CC.

### **1.5 Sistemas Herramientas para el Reconocimiento de Patrones.**

Se entiende por **Sistema Herramienta**: Un complejo de programas (de computadora) orientados a la solución de una familia de problemas, de una o más áreas específicas del conocimiento, tal que el usuario sólo requiere del conocimiento del área específica y además el complejo de programas, no resulta para él una caja negra. Si el sistema es aplicable a cualquier área del conocimiento lo denominaremos Universal [35].

A continuación, se realiza un análisis de los principales sistemas herramientas que existen a nivel nacional e internacional, teniendo en cuenta un conjunto de aspectos, con el objetivo de determinar qué sistema herramienta sería el ideal para su extensión:

- Capacidad de la herramienta para permitir el trabajo simultáneo tanto con variables cualitativas y cuantitativas. Cada uno de estos tipos de variables se debe contar con diversos criterios de comparación, entre dos valores de una misma variable. Se debe permitir para todos los tipos de datos la ausencia de información o valor “?” en su dominio.
- Capacidad para definir dominios en variables descriptivas, y manejo intrínseco de los dominios de datos más comunes (enteros, booleanos, reales, etc.), con posibilidad para su extensión atendiendo a las necesidades propias del problema.
- Capacidad para definir nuevos criterios de comparación y funciones de analogía entre patrones. Brindar al usuario útiles para la creación de funciones de analogía, que permitan la extensión de la herramienta.

- Capacidad de extensión con nuevos algoritmos, para el procesamiento de la información, desarrollados por los usuarios.
- Proveer de utilidades que permitan comparar los resultados de aplicar diferentes algoritmos, sobre un mismo conjunto de datos.
- No debe presuponer un tipo especial de espacio de representación para los datos (métrico, vectorial, euclidiano, booleano, etc.). El espacio dependerá en gran medida del problema a modelar y de los datos que sean necesarios utilizar, sin restricciones en cuanto a tipo.

### **1.5.1 Sistema ACLAPIP**

Es un sistema de programas dirigido a la solución de problemas de clasificación, de diagnóstico y pronóstico de objetos o fenómenos, en los que el valor informativo de cada objeto o estado del fenómeno en estudio, es diferente a los demás y esto se desea tener en cuenta. El sistema brinda posibles variantes de aplicaciones, del modelo de reconocimiento de patrones, basado en el peso informacional de los objetos acorde a la Teoría de Testores [36].

ACLAPIP, consta de un programa principal y 24 subrutinas. Desde el programa principal se realizar una secuencia de llamadas a subrutinas, que realizan las diferentes variantes de cuatro parámetros (Patrón, RSOL, Distancia, Unicidad) esenciales, que caracterizan a los 36 algoritmos que efectúa el sistema.

La importancia de ACLAPIP radica fundamentalmente, en ser la primera herramienta de su tipo, diseñada y construida desde el enfoque lógico- combinatorio. Se trata en efecto de la primera herramienta de software, para el cálculo de testores y de las diversas medidas que de ellos se desprenden, (como la relevancia informacional de objetos y rasgos, en un problema de RP). Sin embargo, como deficiencia más notoria se tiene: la ausencia de mecanismos para definir funciones de semejanza entre patrones. El sistema tiene dos formas de comparación, entre las cuales el usuario puede seleccionar la que mejor se adapte a las condiciones de su problema. No es posible plantear problemas bajo diferentes condiciones de comparación. No presenta mecanismos de comparación para los resultados obtenidos por diferentes algoritmos, aplicados sobre un mismo conjunto de problemas [36].

ACLAPIP es un sistema informático para el RLCP que data de mediados de la década de los ochenta, desde su concepción no fue pensado para permitir la incorporación de nuevas funcionalidades (algoritmos, tipos de variables, funciones de semejanza, criterios de comparación). Los investigadores que hagan uso de la herramienta para solucionar sus problemas prácticos verán muy limitado el espectro de modelos permisibles por la misma.

### **1.5.2 Sistema PROGNOSIS**

PROGNOSIS es un sistema herramienta para expertos en Geociencias, que emplea técnicas avanzadas para la solución de problemas con aprendizaje, aprendizaje parcial y sin aprendizaje. Este sistema está subordinado a una metodología de modelación matemática de fenómenos geofísicos. La herramienta permite profundizar en estos modelos, mejorar el trabajo simultáneo de variables cualitativas y cuantitativas como lo hace el investigador geocientífico, mejorar la calidad del pronóstico, permitir la elaboración de mapas y gráficos [37].

Esta herramienta permite resolver problemas relacionados, con la determinación del pronóstico de terremotos, agrupamiento de zonas geológicas y la determinación de la importancia de cada rasgo.

PROGNOSIS se encuentra desarrollada con el objetivo de solucionar problemas del RLCP como lo haría un geocientífico. Se puede encontrar en ella el análisis comparativo de objetos geológicos, el tratamiento diferenciado de atributos, el trabajo simultáneo de variables cuantitativas y cualitativas, el uso de funciones de semejanzas como eje central de análisis, y admite trabajar con ausencia de información.

PROGNOSIS ha sido empleada en la solución de problemas relacionados con la determinación del pronóstico de terremotos, agrupamiento de zonas geológicas y la determinación de la importancia de cada rasgo en la ocurrencia de fenómenos geofísicos. Resulta necesario señalar en el contexto de la presente investigación que para el cálculo de todos los testores típicos se emplea el algoritmo TB de escala exterior [32].

La herramienta, ofrece un conjunto de funciones interconstruidas de las cuales el usuario puede seleccionar la que será usada en cada problema, pero carece de mecanismos para definir nuevas funciones dinámicamente. Similar ocurre en el caso de los algoritmos o procedimientos, sólo es posible aplicar aquellos interconstruidos en el sistema, no existe forma de que el sistema incorpore algoritmos programados por los usuarios para poder aplicarlos a un conjunto de datos previamente definido [37].

PROGNOSIS es un Sistema Herramienta que tiene su origen a principios de los años noventa. Según sus propios autores es un sistema modular que permitir un fácil versionado (incorporación de nuevas funciones), característica solo disponible a sus desarrolladores. PROGNOSIS está limitado solo al conjunto de aquellos modelos permitidos por él, y que aborden los rasgos a la manera de un geocientífico.

### 1.5.3 Sistema SELECTOR

SELECTOR es un Sistema Herramienta Universal, es decir, un complejo de programas orientados a la solución de problemas de selección de variables, que aparecen en una o más áreas del conocimiento y que para su utilización solo se requiere del dominio de su área específica, además el complejo no constituye una "caja negra" para el usuario. Esto último significa que puede saber lo que el sistema hará con sus datos y así podrá hacer una interpretación real de los resultados [35] [38].

Este sistema posee tres características fundamentales:

- **AUTONOMIA:** Para utilizarlo solamente se requiere tener conocimiento del área de especialidad del usuario, es decir, en Biología, Geología, Medicina, etcétera.
- **TRANSPARENCIA:** Toda la fundamentación matemática del sistema está expuesta con rigor, pero sin tecnicismos, al alcance de especialistas no matemáticos. Esto permite hacer una real interpretación de los resultados que se alcancen, ya que se podrá saber cómo el sistema llega a los mismos.
- **FLEXIBILIDAD:** El sistema permite trabajar con variables cuantitativas de cualquier tipo y con combinaciones de unas y otras. También se pueden resolver los mismos problemas, cuando existe la posibilidad de que un objeto esté en más de una clase o que en la descripción de un objeto, no se conozcan algunos de los valores de las variables que los describen.

Teniendo presente las características antes citadas, se tiene que SELECTOR brinda las posibilidades siguientes: conocer el grado de incidencia de los rasgos (variables, factores, características, etc.) que están presentes en el fenómeno que se estudia; tener un criterio para eliminar variables en el caso que se desee reducir el número de las mismas; tener una medida de los vínculos que existen entre las diferentes variables que se necesite (correlación); tener una medida del grado de representatividad de los objetos en la clase que se ubicó y en particular, conocer los objetos más representativos de cada clase (ideales); ordenar los objetos y (o) las variables que lo describen, según una magnitud que el sistema calcula al efecto.

Aun cuando en su concepción SELECTOR dotaría a los investigadores de un fuerte SHU, sólo resolvía el problema de selección de rasgos y objetos, dejando a un lado el resto de los problemas del RLCP. Se acercaba a todas las propiedades de sistema herramienta descritas y mostraba un medio ambiente de trabajo de menús, diálogos y ayuda al usuario en lenguaje bastante sencillo, pero riguroso (34).

SELECTOR solo permite el trabajo con: los tipos de rasgos, los criterios de comparación, las funciones de semejanzas y los algoritmos definidos en él; limitando la incorporación de nuevas características. Solo posibilita el trabajo con objetos descritos en termino de 255 rasgos, por su antigüedad el sistema solo permite un uso de 256Kb para el manejo de los datos y la ejecución de los algoritmos [38].

#### **1.5.4 Sistema ALISA**

Elaborado por el Instituto de Recursos Minerales de la antigua URSS, ALISA fue el primer sistema que se construyó, con el objetivo de servir de instrumento para resolver todas las tareas planteadas en el RP [39] [37].

Con este sistema se resolvieron muchos problemas de pronóstico y prospección geológica. No obstante, aún dista de ser un Sistema Herramienta. Entre las características más esenciales se pueden señalar que:

- Está construido en base a una metodología, que concibe al especialista matemático, como el único actor posible sobre la selección del mejor algoritmo a aplicar y en la manipulación y elaboración de los datos.
- A pesar de poder solucionar una gran cantidad de problemas de las geociencias, aún adolece de restricciones en cuanto al uso de criterios de comparación que abarquen otras posibilidades, que se acerquen más al modo de tratar la información por el especialista no matemático.
- Al preverse que sea el matemático el que realice la manipulación de los datos y su elaboración, limita el poder experimentador y de creación de los especialistas no matemáticos, crea un sistema rígido, bajo el principio de conocerse todo y que la información inicial es buena.

A pesar de no regirse por los presupuestos del RLCP, se hace valida la mención de la aplicación al ser elaborada en tomando en cuenta las teorías generales del RP. ALISA supone que sea el especialista matemático el que realice la manipulación y elaboración de los datos, limitando el poder experimentador y de creación de los especialistas no matemáticos, crea un sistema rígido, bajo el principio de conocerse todo y que la información inicial es buena. La herramienta no permite el trabajo con variables cualitativas y cuantitativas simultáneamente, ni la ausencia de información. ALISA no soporta la incorporación de nuevas funcionalidades a modo de extensión, a partir de desarrollos propios de los usuarios.

#### **1.5.5 Software 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 [40].

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. Para su explotación no es un requisito poseer conocimientos de programación, solo basta conocer el módulo *Experimenter* y es cuestión de colocar iconos y enlaces para probar cubrimientos. 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 RLCP, por tanto, no es muy útil a los usuarios que hagan uso este enfoque. La incorporación de algoritmos para el RLCP en WEKA deben traer aparejado un cambio en el mismo núcleo de la herramienta, haciendo que pierda parte su objetivo inicial [41][40].

### **1.5.6 Sistema CEPAR**

CEPAR (Entorno Cubano para el Reconocimiento Lógico Combinatorio de Patrones) es un Sistema Herramienta Universal (SHU) desarrollado según las teorías y presupuestos del RLCP. 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 [42].

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, o algoritmos. 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

RLCP. La herramienta para permitir una correcta incorporación de los nuevos desarrollos, que se generen como extensión de la misma, posee interfaces para cada elemento que soporta extensiones.

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 rasgos resulta minúscula, en comparación con los que pueden aparecer en problemas reales del RLCP, por lo que resalta en CEPAR el permitir la extensión según las interfaces definidas de nuevos rasgos por los usuarios, no limitando la herramienta solo al trabajo con los seis rasgos nativos.

Para cada uno de los tipos de rasgos nativos, o implementados, es posible definir dominios de pertenencia con posibilidades de extensión atendiendo a las necesidades propias del especialista. Además, cuenta con diversos criterios de comparación y funciones de semejanzas, que de no cumplir con las necesidades del problema modelado pueden ser extendidos [42].

CEPAR en la actualidad no posee algoritmos que permitan trabajar con grandes volúmenes de información, pero admite que se le puedan agregar nuevos algoritmos para tratar este tipo de problemas, una vez que hayan sido implementados. En sentido general la herramienta cumple con los parámetros descritos al inicio del capítulo y que resultan de interés para la investigación

### **1.5.7 Justificación de la selección**

De las herramientas mencionadas anteriormente, se concluye que CEPAR sería la herramienta seleccionada para el desarrollo e integración de los nuevos algoritmos, teniendo en cuenta el análisis realizado y apoyados por los resultados del análisis de otros sistemas (ver [anexo 3](#)), se puede concluir que:

1. Como la presente investigación se encuentra en el marco de las teorías y presupuestos del Reconocimiento Lógico Combinatorio de Patrones, se

consideran apropiadas las herramientas de este enfoque (ACLAPIP, SELECTOR, PROGNOSIS, CEPAR).

2. Siendo de interés las herramientas que permiten la adición de nuevas funcionalidades para incorporar algoritmos para el cálculo de todos los testores típicos de escala interior, y que además estén basadas en el RLCP (CEPAR).

Se selecciona como herramienta para implementar los algoritmos seleccionados con anterioridad a CEPAR, resulta adecuado realizar otras evaluaciones:

- Las herramientas analizadas del enfoque lógico combinatorio, en su generalidad poseen una antigüedad superior a los 10 años, y no han sido objeto de mejoras para adaptarlas a las nuevas tecnologías.
- WEKA, es comúnmente usada en la solución de algunos de los problemas del RP, y a pesar de que permite la extensión con nuevos algoritmos, la concepción de la herramienta no cuenta con características para permitir la inserción de algoritmos y/o características del RLCP, sin que represente un cambio en su estructura interna.

## **1.6 Herramientas y tecnologías que se utilizan en la propuesta de solución.**

La herramienta CEPAR propone el uso del lenguaje Java y del entorno de desarrollo NetBeans, por esto no se decide realizar una comparación entre los principales entornos de desarrollo que existen, ni el lenguaje de programación a utilizar, al estar definidos en la herramienta.

### **1.6.1 Entorno de desarrollo integrado (IDE).**

**NetBeans** (*en su versión 8.0.2*): es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación, además puede ejecutarse en Windows, OS X, Linux, Solaris y otras plataformas que soportan una Java Virtual Machine compatible. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

También está disponible NetBeans\_Platform; una base modular y extensible usada como estructura de integración para crear grandes aplicaciones de escritorio. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones [43].

## **1.6.2 Lenguaje de programación.**

### **Java 8:**

Es la 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. La nueva versión de Java primero se pone a disposición de los desarrolladores, para dar tiempo suficiente para realizar las pruebas y certificaciones, antes de que esté disponible para su descarga en el sitio web [java.com](http://java.com).

Java ha sido probado, ajustado, ampliado y probado por toda una comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas. Está diseñado para permitir el desarrollo de aplicaciones portátiles, de elevado rendimiento, para el más amplio rango de plataformas informáticas posible. Al poner a disposición de todo el mundo, aplicaciones en entornos heterogéneos, las empresas pueden proporcionar más servicios y mejorar la productividad, las comunicaciones y colaboración del usuario final y reducir drásticamente el costo de propiedad, tanto para aplicaciones de usuario, como de empresa. Java se ha convertido en un valor impagable para los desarrolladores, ya que les permite:

- Escribir software en una plataforma y ejecutarla virtualmente en otra
- Crear programas que se puedan ejecutar en un explorador y acceder a servicios Web disponibles
- Desarrollar aplicaciones de servidor para foros en línea, almacenes, encuestas, procesamiento de formularios HTML y mucho más
- Combinar aplicaciones o servicios que utilizan el lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización
- Escribir aplicaciones potentes y eficaces para teléfonos móviles, procesadores remotos, microcontroladores, módulos inalámbricos, sensores, gateways, productos de consumo y prácticamente cualquier otro dispositivo electrónico [44].

### **1.6.3 Herramienta Case: Visual Paradigm:**

Visual Paradigm es una herramienta CASE (Ingeniería de Software Asistida por Computación por sus siglas en inglés): propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y diseño, hasta la generación de código fuente de los programas y la documentación. Tienen como objetivo fundamental, soportar el ciclo de vida completo del proceso de desarrollo de software, a través de la representación de todo tipo de diagramas. El software de modelado UML contribuye a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación [45].

En el presente capítulo, se abordan los aspectos relacionados con el dominio de la investigación. Se caracterizan los algoritmos de escala interior CT y Fast-CT, utilizados para calcular todos los testores típicos. Se explican los pasos de cada uno de los algoritmos implementados, además de cómo se integran los mismos a la herramienta CEPAR.

## 2.1 Descripción e Implementación.

Los algoritmos de escala interior, construyen el conjunto de TT atendiendo a la estructura interna de la Matriz Básica (MB), teniendo en cuenta la distribución de unos y ceros en dicha matriz. A continuación, se describen los aspectos fundamentales sobre dos de estos algoritmos (CT y Fast-CT), seleccionados en la presente investigación, para su implementación.

### 2.1.1 Algoritmo CT.

A continuación, se realiza una descripción de los principales conceptos que hay que tener en cuenta para desarrollar el algoritmo CT.

**Definición 1:** Sean  $i, j \in I$ . Se dice que  $j$  es una fila aislada de  $i$  respecto a  $k \in J$  si y sólo si  $(a_{ik} = 1) \Rightarrow (a_{jk} = 0)$ .

**Definición 2:** Se llamará a  $r = \{i_1, \dots, i_s\} \subseteq I$  conjunto independiente respecto a  $K = \{k_1, \dots, k_s\} \subseteq J$  si y sólo si se cumple:

$$\forall p \in N; \forall q \in N \left[ 1 \leq p < q \leq s \text{ y } (a_{i_p k_p} = 1) \text{ y } (a_{i_q k_p} = 0) \right].$$

donde  $N$  es el conjunto de los números naturales.

**Definición 3:** Sea  $K = \{k_1, \dots, k_s\} \subseteq J$ , se llamará a  $K$  conjunto completo si y sólo si, se cumplen las siguientes condiciones:

1. Existe un conjunto  $R = \{i_1, \dots, i_s\} \subseteq I$  independiente respecto a  $K$ ;
2.  $\forall i \in I \setminus R, \exists p \in N \left[ 1 \leq p \leq s \text{ y } a_{ik_p} = a_{i_p k_p} = 1 \right]$ .

**Teorema 1:** Para que  $K = \{k_1, \dots, k_s\} \subseteq J$  sea un testor, es suficiente que  $K$  sea un conjunto completo.

**Teorema 2:** Sea  $K = \{k_1, \dots, k_s\} \subseteq J$ . Para que  $K$  sea un testor típico, es necesario y suficiente que  $K$  sea un conjunto completo, y además se cumpla:

$$\forall k \in K, \exists i \in I, \forall r \in K [(a_{ik} = a_{ir} = 1) \Rightarrow (r = k)].$$

**Teorema 3:** Sea  $K = \{k_1, \dots, k_s\} \subseteq J$ . Para que  $K$  sea un testor típico sobre la MB, es necesario y suficiente que  $K$  sea un conjunto completo, cualquiera sea el orden en la MB.

**Algoritmo #1: CT**

**Entrada:** MB.

**Salida:** Lista de testores típicos

**Paso 1:** Ordenar la MB.

**Paso 2:** Inicialización. Se hace  $s = 1; t = 1; I^t = \{i_t\}; K^1 = \{k_1\}$  siendo  $i_1 = 1$

y  $k_1 = \min \{j | a_{ij} = 1\}; i \leq j \leq n$ .

**Paso 3:** Buscar una fila aislada de las filas  $i_1 \dots i_s$  respecto de  $k_1 \dots k_s$ . Si

Existe  $j = \min \{j | a_{ikp} = 0, p = 1, \dots, s\}; i_s < j \leq m$  entonces se toma  $I^t = I^t \cup \{j\}$ .

Si no existe  $j$  que cumpla lo anterior entonces  $K^t$  es un conjunto completo y se va al paso 5.

**Paso 4:** Buscar un elemento unitario en la fila  $j = i_{s+1}$ . Se hace  $K^t = K^t \cup \{k = k_{s+1}\}$ , donde  $k = \min \{j | a_{ij} = 1\}$ ,  $k_s < i \leq n$ . Se fija el valor de  $s = s + 1$  y se regresa al paso 3.

**Paso 5:** Determinar si  $K^t$  es un testor típico. Si lo es entonces se inserta en la lista de testores típicos.

**Paso 6:** Buscar los conjuntos siguientes de  $I^t$  y  $K^t$ .

Si existe  $k = \min \{j | a_{ij} = 1\}$ ,  $k_s < i \leq n$ , se hace  $k_s = k$ , obteniendo así  $K^{t+1} = \{k_1, \dots, k_s = k\}$ ,  $t = t + 1$  y se regresa al paso 3.

**Paso 7:** Reducir los conjuntos siguientes de  $I^t$  y  $K^t$  (el último elemento unitario de la fila  $i_s$  es el  $k_s$ , por ello se regresa a tomar el próximo uno de la  $i_{s-1}$  que se encuentra después de la posición  $k_{s-1}$ ). Si  $s = 1$ , se ha terminado el proceso de búsqueda de los conjuntos completos, en la lista han quedado todos los testores típicos. Si  $s > 1$ , se toma  $I^t = I^t \setminus \{i_s\}; K^t = K^t \setminus \{k_s\}; s = s - 1$  y se regresa al paso 6.

---

A continuación, se muestra un ejemplo del funcionamiento del algoritmo y los testores típicos que generaría el mismo, en cada una de sus iteraciones.

**Ejemplo 1:**

$$MB = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

**Primera iteración**

En una primera iteración, el algoritmo seleccionaría en  $I^1$ , a las filas en las cuales encontró los valores unitarios y en  $K^1$ , guardaría las posiciones de las columnas en las cuales, encontró los valores unitarios y luego comprobó si era un testor típico, determinando en este caso que sí lo era.

$$I^1 = \{1, 3, 4\}$$

$$K^1 = \{2, 6, 5\} \text{ Testor Típico}$$

**Segunda iteración**

En una segunda iteración del algoritmo,  $I^2 = 1,2,3,4$ , serán los valores de las filas donde se obtuvieron los  $K^2 = 4,1,6,5$ , una vez comprobado si es testor típico, se llegó a la conclusión que sí lo es.

$$I^2 = \{1, 2, 3, 4\}$$

$$K^2 = \{4, 1, 6, 5\} \text{ Testor Típico}$$

**Tercera iteración**

$$I^2 = \{1, 2, 3, 4\}$$

$$K^2 = \{4, 2, 6, 5\} \text{ Testor no Típico}$$

**2.1.2 Algoritmo Fast-CT.**

Para la implementación de este algoritmo, fue necesario tener un conocimiento previo del concepto de columnas redundantes, al ser este algoritmo una versión mejorada del algoritmo CT, su objetivo es aumentar la eficiencia del algoritmo eliminando las columnas redundantes.

## Reducción de columnas

Reducción de columnas (**paso 1 del algoritmo**), consiste en formar una nueva matriz básica (reducida), con una columna (función representativa) de cada grupo. Los testores típicos calculados a partir de MB reducida, se llamarán pseudo-testores típicos, porque se obtienen a partir de la MB reducida. La meta de la reducción de columnas se basa en la siguiente proposición.

**Proposición 1.** Sea  $L = [x_1, x_2, \dots, x_s]$  una lista de rasgos,  $x \in L, i = 1 \dots s$ . Si dos rasgos  $x$  y  $x_i$  con sus columnas correspondientes en la MB son iguales, entonces el rasgo  $x$  es excluyente del rasgo  $x_i$  en la lista  $L$ .

**Comprobación.** Como  $x$  y  $x_i \in L$  tienen sus columnas en la MB, todas las filas de  $x$  son las mismas que de  $x_i$ . Por lo tanto, por la Proposición 1,  $x$  es excluyente con  $x_i$  en  $L$ .

De esta proposición, se desprende que dos columnas iguales de la MB, no pueden estar en la misma lista para formar un testor típico, porque si coincide en la MB, los mismos valores para cada fila, el  $x_i$  de la característica hace que la función  $x$  no pueda darle ninguna fila típica en la lista y al mismo tiempo,  $x$  elimina todas las filas típicas de  $x_i$  de la función.

La generación de un testor típico puede hacerse a partir de los pseudo-testores típicos. Si hay dos columnas iguales en MB, llamadas  $C_i, C_j = 1, \dots, s, i \neq j$ , entonces cualquier subconjunto de la característica que genera un testor típico con columna  $C_i$ , también incluyen la columna  $C_j$ . Esto no permite pérdida de testores típicos, una vez que se haya reducido la MB y estos testores típicos se pueden recuperar a través de los pseudo-TT.

El procedimiento para generar el conjunto de testores típicos, de los pseudo-TT, se elabora combinando todos posibles sustituciones de los elementos, que representan a los pseudo-TT, por sus características de pertenencia en la MB. Este proceso tiene un bajo costo computacional.

Ejemplo 2. Sea  $[x_1, x_2 \dots x_7]$ . Las características de la siguiente MB:

$$MB = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Los conjuntos de características con igual número de columnas de esta BM son:

$\{x_1, x_3\}, \{x_4, x_5, x_7\}, \{x_2\}, \{x_6\}$ , que son representadas por las características  $x_{r1}, x_{r2}, x_{r3}$  y  $x_{r4}$  respectivamente para formar el siguiente compuesto reducido por MB  $\{x_{r1} x_{r2}, x_{r3}, x_{r4}\}$

$$MB = \begin{pmatrix} X_{r1} & X_{r2} & X_{r3} & X_{r4} \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Los pseudo-TT generados a partir de esta MB reducidas son:  $[[x_{r1} x_{r2}], [x_{r1} x_{r3}], [x_{r2} x_{r4}]]$ .

Entonces, de  $[x_{r1} x_{r2}]$ , se obtienen los siguientes testores típicos:  $[[x_1, x_2], [x_2, x_3]]$ , de  $[x_{r1} x_{r3}]$ , se obtienen  $[[x_1, x_4], [x_1, x_5], [x_1, x_7], [x_3, x_4], [x_3, x_5], [x_3, x_6]]$ , finalmente de  $[x_{r2} x_{r4}]$  se obtiene, el siguiente testor típico:  $[x_2, x_6]$ .

Como puede observarse, la reducción del número de columnas de la MB, implica una disminución en el tiempo de ejecución de los testores típicos, puesto que se está trabajando con un menor número de columnas. Además, el procedimiento para obtener el testor típico de los pseudo-TT, tiene bajo costo computacional. Esto representa una ventaja siempre que sea posible aplicar este procedimiento.

La reducción de columnas implica, trabajar en un espacio reducido de búsqueda, de  $2^n$  a  $2^{nk}$ ,  $k > 0$ , donde este valor, depende del valor de porcentaje de reducción de columnas obtenidas de la MB. Este hecho no permite que el algoritmo propuesto pueda verificar cuantas combinaciones de columnas son posibles. El algoritmo CT no incorpora este procedimiento. Es importante destacar que el procedimiento de Fast-CT es una extensión del algoritmo CT, mediante la reducción de los procesos de columnas.

Pasos:

1. Reducir columnas redundantes de la matriz básica
2. Ordenar la Matriz Básica.
3. Determinar fila aislada.
4. Determinar conjunto independiente.
5. Hallar conjunto completo.
6. Generar los testores.

---

### Algoritmo #2: Fast-CT

---

**Entrada:** MB.

**Salida:** Lista de testores típicos

**Paso 1:** Eliminar columnas redundantes en la MB.

**Paso 2:** Ordenar la MB.

**Paso 3:** Inicialización. Se hace  $s = 1; t = 1; I^t = \{i_t\}; K^1 = \{k_1\}$  siendo  $i_1 = 1$

y  $k_1 = \min \{j | a_{ij} = 1\}; i \leq j \leq n$ .

**Paso 4:** Buscar una fila aislada de las filas  $i_1 \dots i_s$  respecto de  $k_1 \dots k_s$ .

Si

Existe  $j = \min \{j | a_{ikp} = 0, p = 1, \dots, s\}; i_s < j \leq m$  entonces se toma  $I^t = I^t \cup \{j\}$ .

Si no existe  $j$  que cumpla lo anterior entonces  $K^t$  es un conjunto completo y se va al paso 5.

**Paso 5:** Buscar un elemento unitario en la fila  $j = i_{s+1}$ . Se hace  $K^t = K^t \cup \{k = k_{s+1}\}$ , donde  $k = \min \{j | a_{ij} = 1\}, k_s < i \leq n$ . Se fija el valor de  $s = s + 1$  y se regresa al paso 3.

**Paso 6:** Determinar si  $K^t$  es un pseudo\_testor típico. Si lo es entonces se inserta en la lista de testores típicos.

**Paso 7:** Buscar los conjuntos siguientes de  $I^t$  y  $K^t$ .

Si existe  $k = \min \{j | a_{ij} = 1\}, k_s < i \leq n$  se hace  $k_s = k$ , obteniendo así  $K^{t+1} = \{k_1, \dots, k_s = k\}, t = t + 1$  y se regresa al paso 3.

**Paso 8:** Reducir los conjuntos siguientes de  $I^t$  y  $K^t$  (el último elemento unitario de la fila  $i_s$  es el  $k_s$ , por ello se regresa a tomar el próximo uno de la  $i_{s-1}$  que se encuentra después de la posición  $k_{s-1}$ ). Si  $s = 1$ , se ha terminado el proceso de búsqueda de los conjuntos completos, en la lista han quedado todos los testores típicos. Si  $s > 1$ , se toma  $I^t = I^t \setminus \{i_s\}; K^t = K^t \setminus \{k_s\}; s = s - 1$  y se regresa al paso 6.

**Paso 9:** Convertir los pseudo\_testores típicos en testores típicos.

## 2.2 Integración de los algoritmos en la herramienta

La herramienta CEPAR fue desarrollada sobre la tecnología Java, la misma está compuesta por un conjunto de paquetes de clases, entre los que se pueden encontrar el paquete *cepar.core*, en el cual se encuentran implementadas todas las funcionalidades que permiten

realizar un correcto modelado del problema, según el Reconocimiento Lógico Combinatorio de Patrones. Este paquete además posee todas las interfaces, clases y abstracciones necesarias, que garantizan la comunicación entre cada uno de los paquetes del reconocimiento patrones. Otro de los paquetes presente en la herramienta es, *cepar.featuresSelection*, el cual cuenta con las características necesarias para la resolución de problemas de selección de rasgos y/o objetos, y en el cual serán integrados cada uno de los algoritmos. La interfaz *IOutputTypicalTestor*, permite a la herramienta generar los testores típicos calculados por los algoritmos. La interfaz *ICalculateTypicalTestor*, contiene dos métodos que son imprescindibles para la implementación de los algoritmos, uno de ellos es (*Inicializar*), el cual recibe la matriz básica y el otro es *Execute*, el cual recibe un dato de tipo *IOutputTypicalTestor* y se realiza el cálculo de los Testores Típicos.

En las figuras 3 y 4 se muestran los diagramas de clases, resultados de integrar los algoritmos a la herramienta CEPAR.

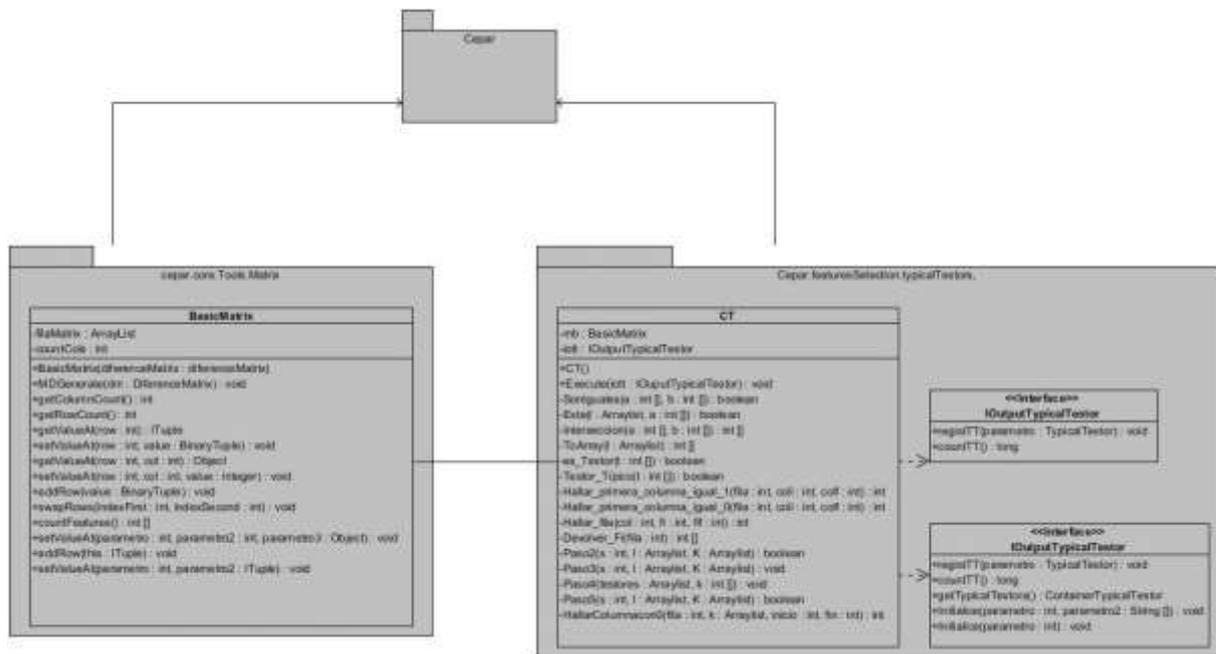


Figura 3 Diagrama de clases del algoritmo CT.

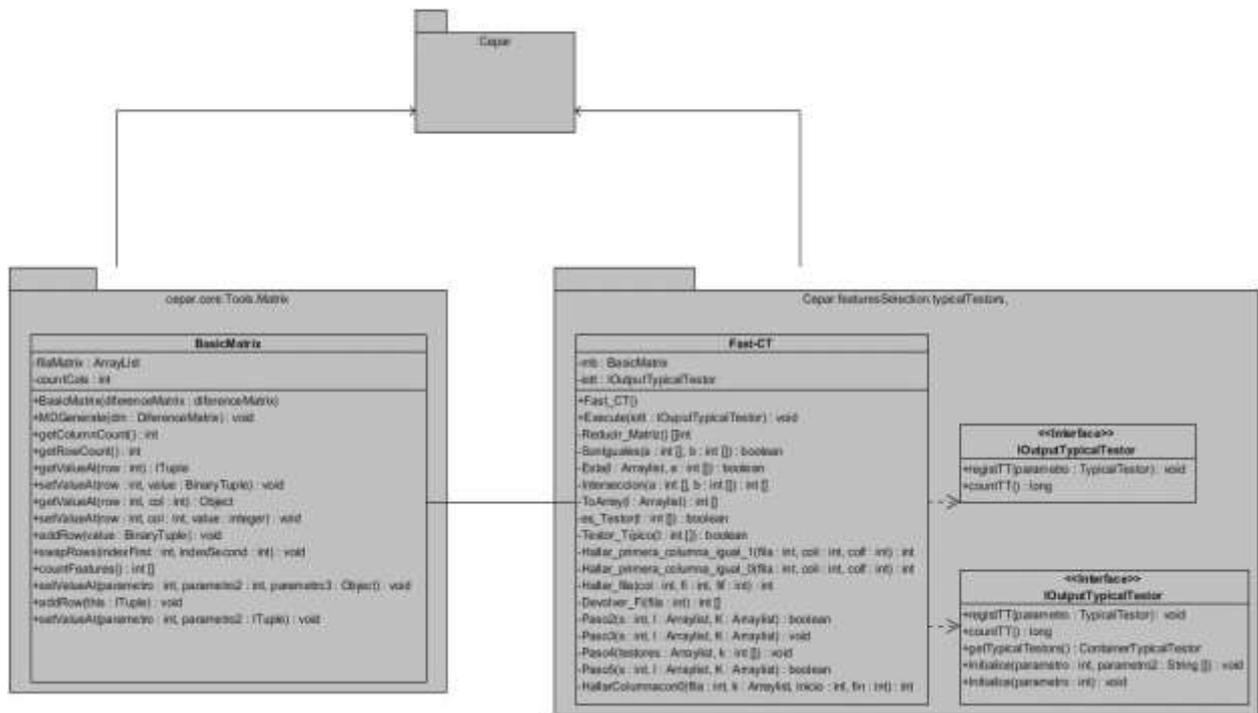


Figura 4 Diagrama de clases del algoritmo Fast-CT.

A continuación, se muestran dos imágenes, la primera indica el estado inicial de la herramienta sin los algoritmos integrados y una segunda imagen indica, como quedan los algoritmos CT y Fast-CT integrados a la misma, exactamente en el paquete (*TypicalTestor*).

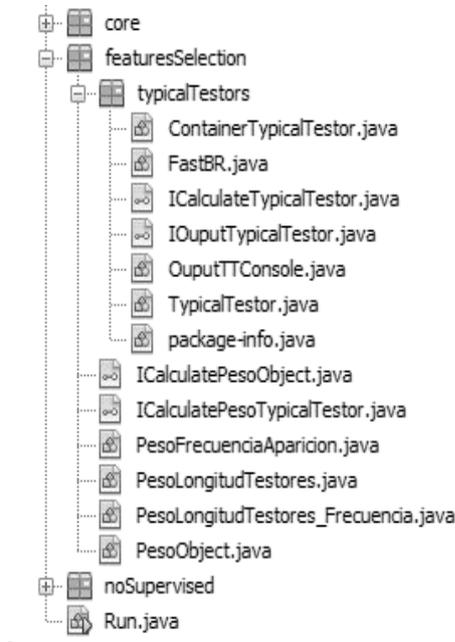


Figura 5 Paquetes de CEPAR.

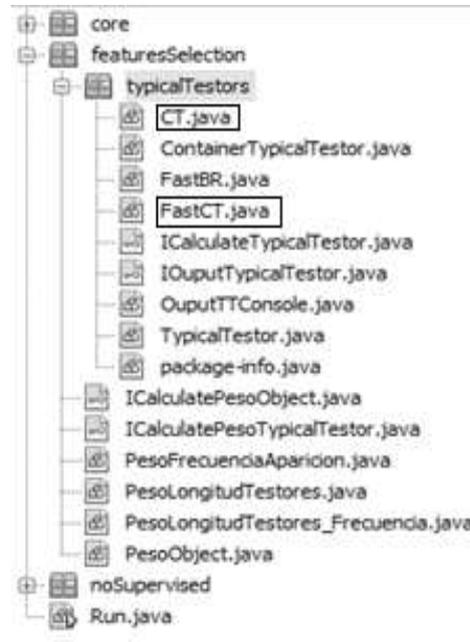
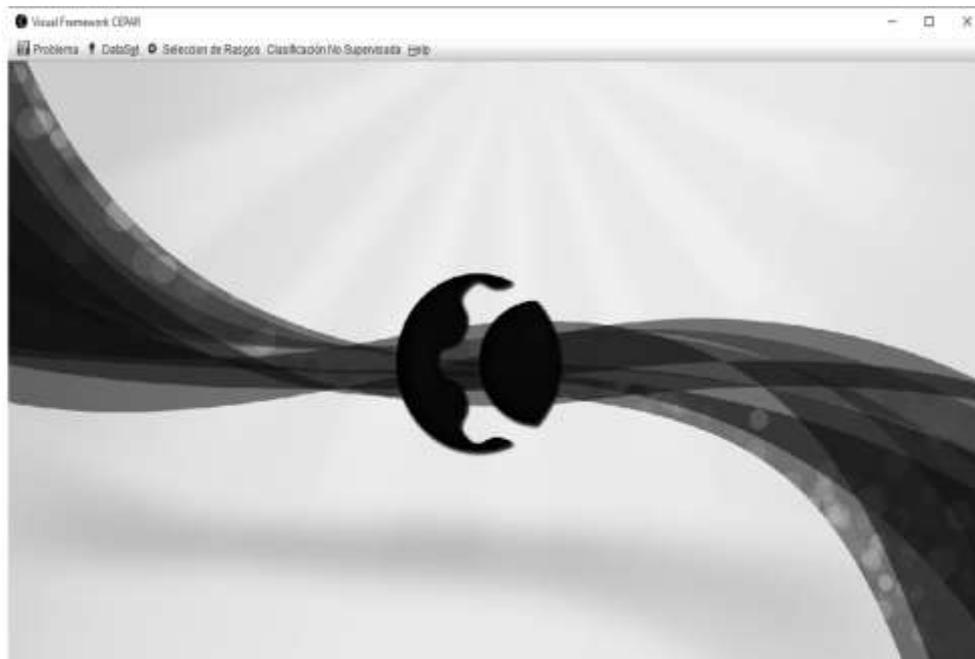


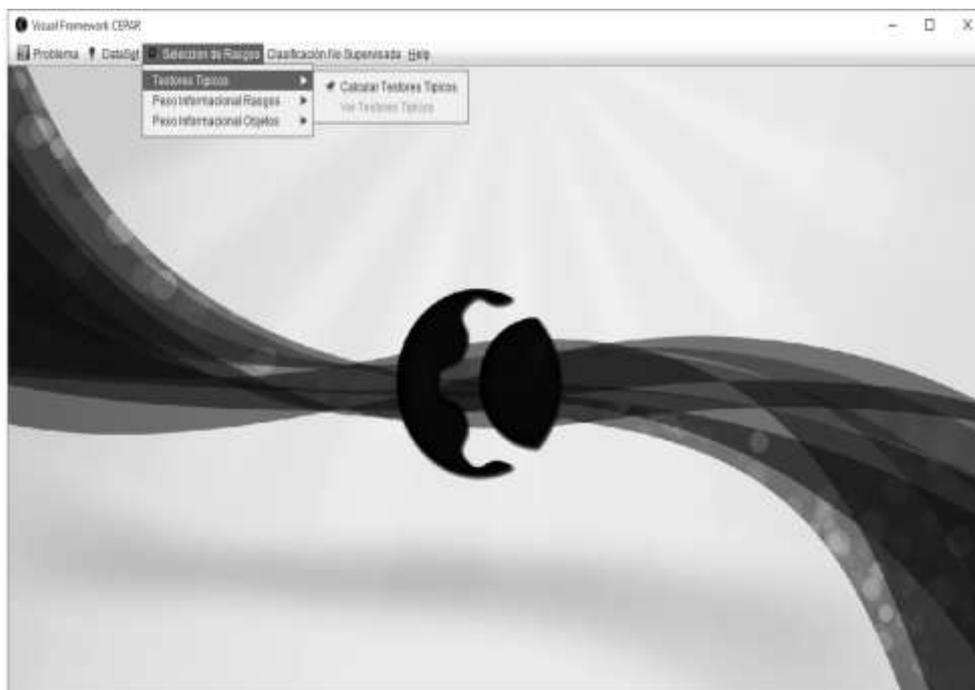
Figura 6 Paquetes de CEPAR con los algoritmos integrados.

A continuación, se muestra el funcionamiento básico de la herramienta CEPAR, para realizar el cálculo de los testores típicos.



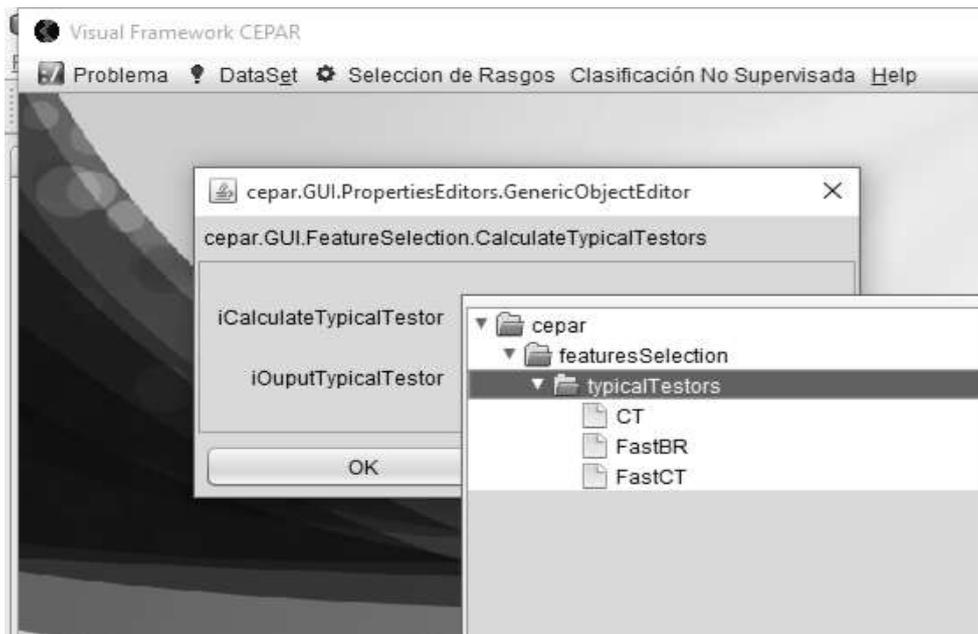
**Figura 7 Herramienta CEPAR**

Esta es la ventana principal de la herramienta CEPAR, la cual está compuesta por un menú el cual brinda la posibilidad de realizar varias operaciones entre ellas calcular testores típicos.



**Figura 8 Menú de selección cálculo de testores típicos.**

En esta ventana es donde se selecciona, testores típicos, el cual permitirá escoger el algoritmo por el cual se desea realizar el cálculo de los testores típicos.



**Figura 9 Seleccionar algoritmo para el cálculo de los testores típicos.**

En esta ventana es donde se seleccionará el algoritmo, por el cual desea realizar el cálculo de los testores típicos, en este caso se seleccionará el algoritmo CT.



**Figura 10 Resultados del algoritmo seleccionado.**

En esta consola es donde se muestran todos los testores típicos, generados por el algoritmo seleccionado, en este caso generó los testores típicos [ 0 ] y [3 , 2], para una base de datos la cual ya había sido insertada en la herramienta.

En el presente capítulo, para evaluar la propuesta de solución de la investigación, se utilizaron varios métodos, los cuales serán aplicados y explicados durante el transcurso del capítulo.

### 3.1 Proceso de validación de la solución

Para la validar los resultados de la investigación, se hizo uso de 3 tipos de técnicas, las cuales son:

- El método IADOV, se realizará una entrevista a un grupo de personas con el objetivo de determinar el nivel de satisfacción de los usuarios, respecto a la propuesta de solución de la investigación.
- El método Delphi, en el cual se realizará una encuesta a un grupo de expertos en el Reconocimiento de Patrones, con el objetivo de validar la propuesta de solución de investigación.
- Desarrollo de casos de estudio: a partir de base de datos internacionales, permitirá comparar los resultados de los algoritmos determinando que existe correspondencia en los testores típicos calculados.

En el siguiente diagrama se muestran los diferentes procesos de validación que se realizaron en la investigación.



Figura 11: Procesos de validación

## 3.2 Validación por expertos

### 3.2.1 Método Delphi

La técnica Delphi [46], es una técnica utilizada para obtener un consenso de opinión sobre un problema complejo, a partir de un grupo de personas especialistas en un área determinada. El uso de ésta técnica permite la obtención de predicciones sin influencia de los individuos dominantes del grupo debido al anonimato de los participantes, ni la presión del grupo hacia la conformidad evitada mediante una retroalimentación controlada por el conductor de la técnica que hace posible la transmisión de información entre expertos libre de ruidos.

A continuación se muestran las características que definen y configuran al método Delphi como una técnica grupal relevante:

**Iteratividad:** Los participantes en el proceso tienen que emitir su opinión en más de una ocasión, a través de rondas sucesivas, a través de las cuales las estimaciones realizadas por ellos tienden a converger, por lo que el proceso finaliza en el momento en el que las opiniones se estabilizan. De esta manera se ofrece al participante la posibilidad de reflexionar y reconsiderar su postura.

**Mantenimiento del anonimato de los participantes:** Esta característica permite eliminar alguna de las causas que inhiben a los participantes como es, el efecto negativo que ejercen algunos participantes dominantes del grupo para la comunicación efectiva. Por ello, ningún miembro del grupo de participantes debe conocer las respuestas del resto de participantes, lo cual puede implicar incluso, si es necesario, que los participantes no conozcan quienes son los otros componentes del grupo. Para lograr el anonimato y evitar la confrontación directa se hace uso de cuestionarios y para conectar a los participantes con el investigador se hace uso de comunicación asistida por ordenador o incluso teléfono o fax. En cualquier caso, el control de la comunicación está en manos del investigador, sin darse nunca interacción directa entre los expertos participantes.

**Retroacción o retroalimentación controlada:** El método Delphi promueve la retroacción solicitándola, a veces de forma expresa en cada ronda o previamente al inicio de la siguiente. Además, mantiene la retroalimentación, ya que antes de comenzar cada ronda, se hace conocer a los participantes la posición general del grupo frente al problema a analizar y, con frecuencia, también de las aportaciones o sugerencias significativas de algún experto desvinculado de quien la realizó. La filtración o control de la comunicación entre los expertos por parte del coordinador tiene como finalidad evitar la aparición de “ruidos”, como información no relevante, redundante, e incluso errónea, o las influencias negativas.

**Respuesta estadística de grupo:** En los casos en los que el grupo de expertos es requerido para realizar una estimación numérica, la respuesta estadística del grupo viene caracterizada generalmente por la mediana de las respuestas individuales.

Según Landeta [47] un experto es “aquel individuo que su situación y sus recursos personales le posibilitan a contribuir positivamente a la consecución del objetivo que ha motivado la iniciación del trabajo”.

Respecto al número de expertos a participar, sólo los estudios de Landeta [47] señalan que el número de expertos a participar debe ser mayor o igual a 5 (el error disminuye exponencialmente por cada experto añadido) y no mayor de 30 (la mejora de previsión es muy pequeña y no compensa costos y tiempo).

En el presente trabajo, el método Delphi ha sido utilizado con el objetivo de validar la propuesta de solución del problema planteado, ya que resulta una preocupación para los investigadores de las teorías del RLCP, no poder realizar la selección del algoritmo más adecuado para su investigación, al encontrarse dispersos entre las herramientas. De igual forma se pretende utilizar el método para comprobar la pertinencia de la propuesta de solución.

### 3.2.2 Proceso de selección de los expertos

Para la selección del grupo de expertos a los cuales se les realizarán la encuesta, el primer paso es establecer como criterio fundamental de selección, la competencia de los candidatos en el área del conocimiento en que se inserta la investigación, basado en su currículo personal. Se identificaron 15 posibles candidatos a seleccionar, de los cuales se descartaron 7, por falta de disposición. Quedando en un total de 8 expertos, como segundo paso, se realiza una valoración sobre el nivel de experiencia que poseen los mismos, evaluando de esta forma los niveles de conocimientos que poseen sobre la materia.

Para determinar el nivel de experiencia que poseen los expertos se realizarán dos preguntas, una pregunta de autoevaluación, de los niveles de información y argumentación que tienen sobre el tema en cuestión. En esta pregunta, se les pide a los expertos que marquen con una X, en una escala creciente del 1 al 10, el valor que se corresponde con el grado de conocimiento o información que tienen sobre el tema.

Expertos	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										

5										
6										
7										
8										

**Tabla 2 Procesamiento del grado del conocimiento sobre el tema.**

Esta primera pregunta permite determinar el Coeficiente de Conocimiento ( $K_c$ ) de cada uno de los expertos, a través de la siguiente fórmula:

$$K_c = n(0.1) = \frac{n}{10}$$

### Ecuación 3 Coeficiente de Conocimiento

Donde:

$K_c$ : Coeficiente de Conocimiento o Información

n: Rango seleccionado por el experto

La segunda pregunta pretende valorar un grupo de aspectos que pueden influir en el nivel de argumentación o fundamentación del tema a estudiar.

Fuentes de argumentación o fundamentación.		Alto	Medio	Bajo
1	Experiencia Teórica			
2	Su experiencia práctica			
3	Bibliografía nacional consultada			
4	Bibliografía internacional consultada			
5	Su conocimiento del estado del problema			
6	Su intuición			

**Tabla 3 Patrón para la determinación del nivel de argumentación.**

Se determinan los aspectos de mayor influencia, a partir de estos valores reflejados por cada experto en la tabla, y se contrastan con los valores de una tabla patrón:

Fuentes de argumentación o fundamentación.		Alto	Medio	Bajo
1	Experiencia Teórica	0.30	0.20	0.10
2	Su experiencia práctica	0.50	0.40	0.20
3	Bibliografía nacional consultada	0.05	0.05	0.05
4	Bibliografía internacional consultada	0.05	0.05	0.05
5	Su conocimiento del estado del problema	0.05	0.05	0.05
6	Su institución	0.05	0.05	0.05

**Tabla 4 Patrón para la determinación del nivel de argumentación.**

Los aspectos que influyen sobre el nivel de argumentación o fundamentación del tema a estudiar permiten calcular el Coeficiente de Argumentación ( $K_a$ ) de cada experto:

$$K_a = \sum n_i = (n_1 + n_2 + n_3 + n_4 + n_5 + n_6)$$

#### Ecuación 4 Coeficiente de Argumentación

Donde:

$K_a$ : Coeficiente de Argumentación

$n_i$ : Valor correspondiente a la fuente de argumentación  $i$  (1 hasta 6)

Una vez obtenido los valores del Coeficiente de Conocimiento ( $K_c$ ) y el Coeficiente de Argumentación ( $K_a$ ), se procede a obtener el valor del Coeficiente de Competencia ( $K$ ), que finalmente es el coeficiente, que determina los expertos que deben tomarse en consideración para trabajar en esta investigación. Este coeficiente ( $K$ ) se calcula de la siguiente forma

$$K = 0.5(K_c + K_a) = \frac{K_c + K_a}{2}$$

#### Ecuación 5 Coeficiente de competencia.

Donde:

$K$  : Coeficiente de Competencia

$K_c$  : Coeficiente de Conocimiento

$K_a$ : Coeficiente de Argumentación

Los resultados se analizan de la siguiente forma para cada experto:

$0,8 \leq K \leq 1,0$  Coeficiente de Competencia Alto

$0,5 \leq K < 0,8$  Coeficiente de Competencia Medio

$K < 0,5$  Coeficiente de Competencia Bajo

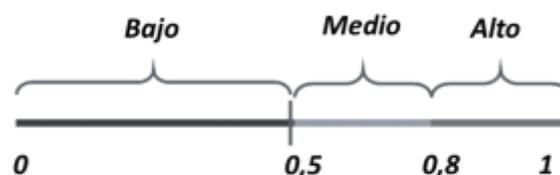


Figura 12 Coeficiente de competencia.

Atendiendo a las características de los expertos, a partir del Coeficiente de Competencia Calculado (ver tabla 5), se determinó realizar la selección de aquellos expertos que tuviesen resultados satisfactorios de desempeño, tomándose aquellos que poseen un valor de coeficiente de competencia (alto), los cuales garanticen estudiar, analizar y dar criterios válidos y confiables, sobre el trabajo elaborado. Quedando finalmente seleccionados un panel de 5 expertos (ver tabla 6).

Expertos	$K_c$	$K_a$	$K$	Valoración
1	0.6	0.8	0.7	Medio
2	1	0.8	0.9	Alto
3	1	0.9	0.95	Alto
4	0.6	0.7	0.65	Medio
5	0.7	0.5	0.6	Medio
6	0.9	0.8	0.85	Alto
7	1	0.9	0.95	Alto
8	0.8	0.8	0.8	Alto

**Tabla 5 Resultados del procesamiento para la determinación del coeficiente de competencia de expertos.**

Nº	Formación académica, científica y experiencias de trabajo en el tema			
	Conocimientos sobre el Reconocimiento Lógico Combinatorio de Patrones	Calificación profesional	Categoría docente	Años de experiencia
2	Si	<i>Máster</i>	<i>Profesor Asistente</i>	8
3	Si	<i>Máster</i>	<i>Profesor Asistente</i>	13
6	Si	Ingeniero	<i>Profesor Asistente</i>	9
7	Si	Doctor	<i>Profesor Titular</i>	30
8	Si	Ingeniero	-	3

**Tabla 6 Caracterización de los expertos que participaron en la encuesta.**

### 3.2.3 Elaboración del cuestionario

Después de haber seleccionado los expertos a los cuales se le realizarán la encuesta, se prosigue a confeccionar las preguntas que integrarán la boleta. La misma está conformada por un total de 5 preguntas, en las cuales se abordarán temas relacionados al problema planteado, la propuesta de solución y los beneficios de la propuesta planteada. El cuestionario (ver [anexo 1](#)) fue desarrollado de forma tal que las respuestas se categorizan de la siguiente forma: (Muy adecuado (A1), Bastante adecuado (A2), Adecuado (A3), poco adecuado (A4) y No adecuado (A5)), todos son equivalentes a un peso de 5, 4, 3, 2,1 respectivamente.

### 3.2.4 Análisis y procesamiento de los resultados

**Establecimiento de la concordancia de los expertos mediante el coeficiente de Kendall.** En la prueba estadística el Coeficiente de Concordancia de Kendall (W), se ofrece el valor que posibilita decidir el nivel de concordancia entre los expertos o especialistas en nuestro caso. Por lo que un perfecto acuerdo entre ellos dará mayor validez a la propuesta. Se muestra a continuación la tabla que refleja la puntuación de las respuestas, en base a 5 puntos, dadas por los especialistas encuestados.

	E1	E2	E3	E4	E5	Rj
Pregunta 1	4	5	5	5	1	<b>20</b>
Pregunta 2	5	5	4	5	5	<b>24</b>
Pregunta 3	5	5	4	5	5	<b>24</b>
Pregunta 4	5	5	5	5	1	<b>21</b>
Pregunta 5	5	5	5	5	4	<b>24</b>
						<b>113</b>

**Tabla 7 Resultados de la encuesta.**

Una vez que se haya establecido la puntuación de las respuestas dadas por los expertos encuestados, se hace uso de la fórmula para el cálculo del Coeficiente Kendall la cual plantea lo siguiente:

$$W = \frac{12s}{k^2(N^3 - N)}$$

#### **Ecuación 6 Coeficiente de Kendall**

- K (5) es el número de expertos que intervienen en el proceso de validación.
- N (5) cantidad de aspectos a validar.
- S es la suma de los cuadrados de las desviaciones y se calcula de la siguiente forma:

$$S = \sum_{j-i}^n (R_j - \bar{R}_j)^2$$

- $R_j$  es la suma de los rasgos asignados a cada pregunta por parte de los expertos.
- $\bar{R}_j$  es la media de los rasgos y se determina mediante la fórmula:

$$\bar{R}_j = \frac{\sum_{j-i}^n R_j}{N}$$

Luego se procede con el cálculo de Chi-Cuadrado:

$$X^2 = K(N - 1)W$$

### Ecuación 7 Chi-Cuadrado.

El Chi cuadrado calculado se compara con el de las tablas estadísticas. Este se compara con el de la tabla inversa de la función de distribución de la variable Chi-Cuadrado, con una probabilidad de error de 0,01.

Si  $x_{real}^2 < x_{(\alpha, N-1)}^2$  entonces existe concordancia entre los expertos.

$$\bar{R}_j = 32.2$$

$$S = (20 - 22.6)^2 + (24 - 22.6)^2 + (24 - 22.6)^2 + (21 - 22.6)^2 + (24 - 22.6)^2$$

$$S = 15.2$$

$$W = \frac{12(15.2)}{5^2(5^3 - 5)}$$

$$W = \frac{12(15.2)}{25(125 - 5)}$$

$$W = \frac{182.4}{3000}$$

$$W = 0.060$$

El valor obtenido expresa el grado de concordancia entre los 5 expertos, al dar un orden evaluativo de las preguntas sometidas a valoración.

Luego se procede a realizar el cálculo de Chi-cuadrado, con el fin de determinar si existe concordancia entre los expertos encuestados.

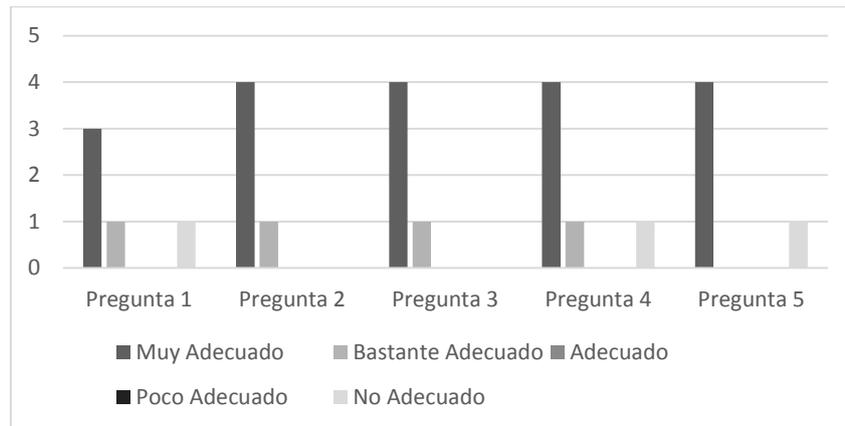
$$X^2 = K(N - 1)W$$

$$X^2 = 5(5 - 1) * 0.060 = 1.2$$

El Chi-cuadrado calculado se compara con el de las tablas estadísticas. Este se compara con el de la tabla inversa, de la función de distribución de la variable Chi-Cuadrado, con una posibilidad de error de 0.01.

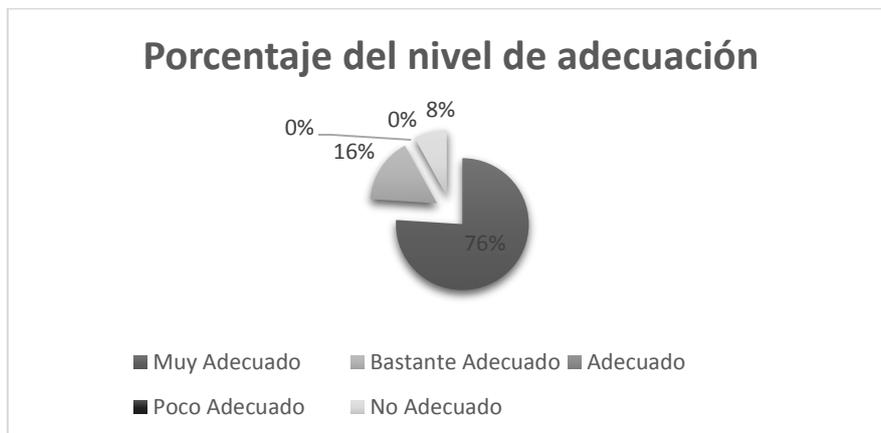
Si  $x_{real}^2 < x_{(\alpha, N-1)}^2$  entonces hay concordancia entre los especialistas  $x_{real}^2 = 1.2$  es menor que  $x_{(0.01, 4)}^2 = 13.27$ . Lo que corrobora, el cumplimiento de la comparación y se puede decir que existe concordancia entre los expertos encuestados, lo que quiere decir que están de acuerdo con la existencia de una herramienta con varios algoritmos para el cálculo de los testores típicos.

A continuación se muestra una gráfica los resultados de la encuesta realizada.



**Figura 13 Gráfica de adecuación de preguntas.**

Una vez finalizado el proceso de validación de cada uno de los datos, se puede llegar a la conclusión, que el 16% de las respuestas fueron evaluadas de Bastante adecuado, el 0% de Adecuado, 0% de Poco adecuada ,8% de No adecuada y el 76% de Muy adecuado, viéndose una clara superioridad de respuestas de muy adecuado. En la siguiente gráfica se puede observar mejor:



**Figura 14 Porcentaje de nivel de adecuado.**

### 3.3 Validación de los algoritmos

El presente epígrafe tiene el objetivo de comprobar el correcto funcionamiento de los algoritmos desarrollados y que la incorporación de varios algoritmos para el cálculo de todos los testores típicos en una herramienta, facilita la selección del mejor algoritmo para la resolución de un problema dado. Para realizar el cálculo de todos los Testores Típicos, con la herramienta seleccionada (CEPAR), una vez que hayan sido integrados los algoritmos a la herramienta.

Para garantizar una comparación equitativa todas las pruebas se llevan a cabo en un procesador Intel Pentium Dual Core a 2,4 GHz, con 4 GB de RAM. Para esta comparación se utilizó tres conjuntos de datos obtenidos a partir de la UCI Machine Learning <sup>1</sup>Repository. Para cada uno, se generaron las matrices básicas relativas al resultado de procesarlas tomando con:

- Criterios de comparación por rasgo (la igualdad estricta):

$$\delta X_i(O_1, O_2) = \begin{cases} 1 \text{ si } X_i(O_1) = X_i(O_2) \\ 0 \text{ en cualquier otro caso} \end{cases}$$

- Función de semejanza entre objetos (la Suma Normalizada):

$$\Gamma(O_i, O_j) = \frac{\sum_{l=0}^n \delta X_l(O_i, O_j)}{n}$$

#### Ecuación 8 Función de semejanza

- Umbral de semejanza:

$$\beta_0 = \frac{1}{m} \sum_{i=1}^m \max_{i \neq j=1, \dots, n} \{ \Gamma(O_i, O_j) \}$$

#### Ecuación 9 Umbral de semejanza

Todas las configuraciones antes expuestas para la generación de la matriz básica que sirve de entrada al cálculo de todos los testores típicos, se encuentran implementadas en la herramienta CEPAR.

Juego de Datos	Clases	MB	NTT	CT	Fast-CT
Zoo(101 x 17)	24	17x17	13	11593 ms	12593 ms
Mushroom (8124 x 22)	2438	13x22	4	109441ms	100364ms
Flags (194x 30)	74	1396x30	101207	+2hr 30min	+2hr 30min

**Tabla 8 Resultados de los algoritmos sobre un mismo juego de datos.**

Los resultados de las distintas ejecuciones de los algoritmos sobre un mismo juego de datos, permiten determinar que algoritmo es más eficiente en cada caso. Lo que garantiza llegar a conclusiones reales, sobre que algoritmo es mejor usar ante los distintos juegos de datos.

<sup>1</sup> <https://archive.ics.uci.edu/ml/index.html>

### 3.4 Validación de satisfacción de usuarios potenciales

#### 3.4.1 Aplicación de la técnica ladov

Se aplicó la técnica **ladov** con el fin de conocer el grado de satisfacción de potenciales usuarios respecto a la propuesta. Esta técnica se basa en la aplicación de un cuestionario (ver [anexo 2](#)) con al menos tres preguntas cerradas y dos abiertas. El cuadro **lógico de ladov** (ver tabla 9), establece la relación entre las tres preguntas cerradas la cuales están dirigidas a conocer la satisfacción respecto al central que motiva la encuesta [48].

La **técnica de ladov** fue creada por la profesión de carreras pedagógicas para establecer el nivel de satisfacción de un grupo de personas. Luego algunos autores han modificado la técnica en parte y aplicado para valorar la satisfacción en múltiples campos y como parte de diagnósticos y validaciones en diferentes investigaciones. En la presente investigación se hace uso de esta técnica específicamente, para medir la satisfacción de potenciales usuarios respecto a la propuesta.

La técnica de ladov constituye una vía indirecta para estudiar la satisfacción de las personas a la hora de desempeñar ciertas actividades. Se basa en el analizar un cuestionario el cual tiene una estructura interna determinada, la cual sigue una relación entre tres preguntas cerradas y el análisis posterior de dos cuestiones abiertas. La relación entre las preguntas cerradas se establece a través del denominado "Cuadro Lógico de ladov".

En la **Tabla** se muestra el Cuadro lógico de V.A. IADOV, donde se realiza la triangulación de las preguntas directas realizadas.

	¿Si usted fuera a realizar un nuevo proyecto preferiría no utilizar la herramienta propuesta para llevar a cabo el cálculo de todos los Testores Típicos?								
	No			No sé			Sí		
¿Satisface sus necesidades como (estudiante   investigador   docente) una herramienta con varios algoritmos para el cálculo de todos los Testores Típicos?	¿Si usted necesitara desarrollar una aplicación en la que se lleve a cabo el cálculo de todos los Testores Típicos usaría esta propuesta?								
	Sí	No sé	No	Sí	No	No	Sí	No sé	No
Me satisface mucho	1	2	6	2	2	6	6	6	6

No me satisface tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me satisface	6	3	6	3	4	4	3	4	4
No me satisface	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

**Tabla 9 Cuadro lógico ladov.**

El número resultante de la interrelación de las tres preguntas indica la posición de cada sujeto en la escala de satisfacción siguiente:

**La escala de satisfacción está compuesta de la siguiente forma.**

<b><i>Escala de satisfacción</i></b>
1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

**Tabla 10 Escala de satisfacción.**

La escala anterior es usada para determinar el Índice de Satisfacción Grupal (ISG), el cual se calcula de la siguiente forma:

**La satisfacción grupal se calcula por la fórmula siguiente:**

$$ISG = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

**Ecuación 10 Índice de satisfacción grupal**

En la fórmula anteriormente representada, N el número total de sujetos que participan en la encuesta, la variable A representa a los encuestados con valor 1, B por los de valor 2, C por los de valor 3 y 6, D por lo de valor 4, E para los de valor 5 [49].

El índice de satisfacción grupal (ISG) da una medida de la satisfacción de los encuestados mediante las siguientes categorías:

- Insatisfacción ( $-1 \leq ISG \leq -0,5$ ).
- Contradictorio ( $-0,5 < ISG < 0,5$ ).
- Satisfacción ( $0,5 \leq ISG \leq 1$ ).



Figura 15: Intervalos de satisfacción grupal.

### 3.4.2 Análisis de los resultados

El cuestionario (ver [anexo 2](#)) fue aplicado a un total de 30 personas, de las cuales 3 no son procesadas al no poseer conocimientos sobre el Reconocimiento Lógico Combinatorio de Patrones, quedando finalmente una muestra de 27 personas. De las cuales seis eran especialistas o profesores, y el resto estudiantes que cursaron (o cursan), la asignatura optativa Reconocimiento de Patrones en la Facultad 2, de la Universidad de las Ciencias Informáticas. Es necesario destacar que los estudiantes que se matricularon en la asignatura Reconocimiento de Patrones, en el segundo semestre del curso 2015-2016, desarrollaron su proyecto final haciendo uso de la herramienta CEPAR.

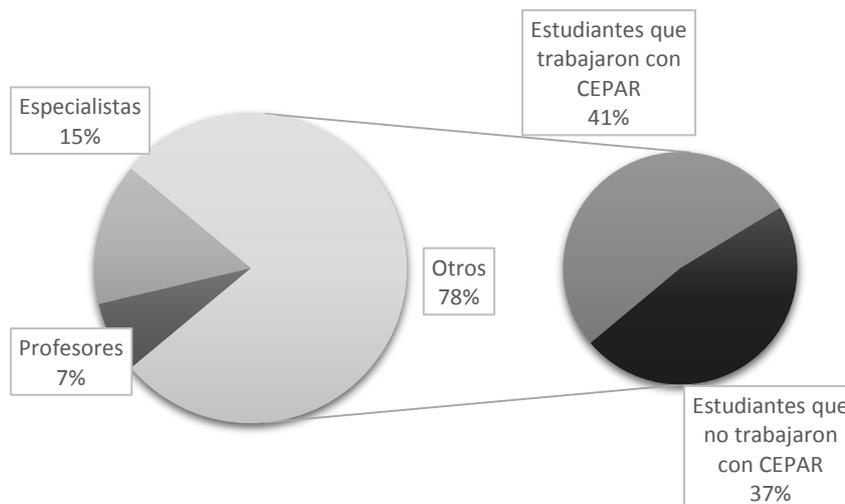
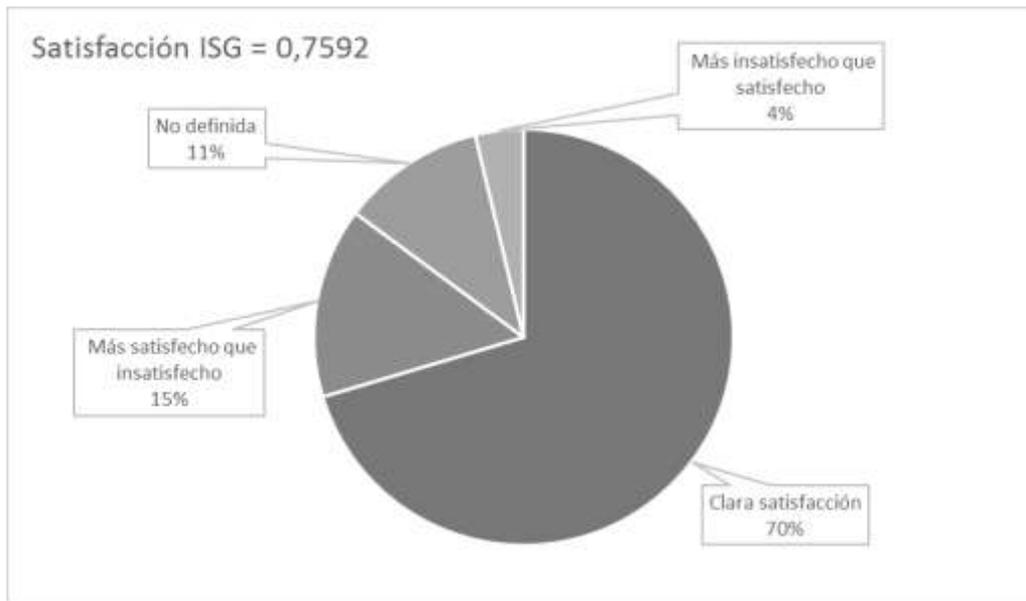


Figura 16: Características de los encuestados de ladov.

El índice grupal (ISG) arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción de los encuestados; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción. En este caso el valor del ISG fue de 0.7592 lo que indica satisfacción con respecto a la propuesta. El resultado del nivel de satisfacción se muestra en la siguiente figura:



**Figura 17: Resultados de la aplicación de la técnica de ladov y valor de índice de satisfacción grupal (ISG).**

### **3.5 Resultados de las pruebas y validaciones**

Una vez finalizada la ejecución de los algoritmos CT y Fast-CT sobre las base datos Zoo, Mushroom y Flags se llegó a la conclusión que generaban la misma cantidad de testores típicos, además se determinó el algoritmo Fast-CT es más eficiente en base datos en las cuales pueda reducir la matriz básica, en caso de no ser posible como en la base de datos Mushroom obtiene tiempos de ejecución mayor que el algoritmo CT. Esto permitiría realizar comparaciones entre los algoritmos a los investigadores, lo que les posibilitaría realizar la selección del más adecuado para su investigación.

---

## CONCLUSIONES

---

La presente investigación que comenzó con el objetivo de obtener una herramienta para apoyar la selección y comparación de algoritmos para el cálculo de todos los testores típicos. Con dicho propósito se han abordado temas a fines relacionados con el RLCP, la Teoría de Testores, los algoritmos de escala interior para el cálculo de todos los Testores Típicos y los Sistemas Herramientas existentes para el RLCP; concluyendo que:

- El estudio de los principales algoritmos de escala interior que existen, permitió identificar los algoritmos a desarrollar en la presente investigación, dando como resultados a los algoritmos CT y Fast-CT.
- Se implementaron los algoritmos de escala interior CT y Fast-CT, teniendo en cuenta las características del sistema herramienta CEPAR.
- Se identificaron y seleccionaron las bases de datos para evaluar los resultados brindados por los algoritmos, comprobándose los mismos.
- El método ladov para evaluar el nivel de satisfacción de los clientes, respecto a la propuesta de solución, da muestra de que los clientes se encuentran satisfechos con el desarrollo de un Sistema Herramienta que posee varios algoritmos para el cálculo de los testores típicos.
- El método Delphi fue empleado para validar la propuesta de solución, así como el problema planteado, a partir de la consulta a expertos, obteniéndose como resultados satisfactorios.

---

## RECOMENDACIONES

---

- Se recomienda la incorporación de otros algoritmos de escala interior para el cálculo de todos los testores típicos en la herramienta CEPAR, destacando entre ellos el CC y el YYC.
- Emplear la herramienta sobre la cual se desarrolla la investigación en todas las facultades de la universidad en la asignatura optativa de Reconocimiento de Patrones.
- Desarrollar mecanismos en la herramienta CEPAR que permitan determinar la relevancia de cada uno de los testores típicos calculados.

---

## REFERENCIAS

---

- [1] J. E. Guzmán-Trampe, «Extensión de un lenguaje formal (LCARS) para especificar problemas de clasificación y de selección de rasgos, mediante la construcción de un intérprete optimizado», Instituto Politécnico Nacional, 2009.
- [2] J. Ruiz-Shulcloper, M. García-Borroto, J. A. Carrasco-Ochoa, José Francisco Martínez-Trinidad, y M. Medina-Perez, «An Efficient Algorithm for Mining Discriminative Regularities and its Application in Supervised Classification. Pattern Recognition», 2010.
- [3] L. Carvajal-Hernández, JJ Sánchez-Fernández, L. Villa-Vargas, J. A. Carrasco-Ochoa, y José Francisco Martínez-Trinidad, «Water quality assessment in shrimp culture using an analytical hierarchical process.», *Ecol. Indic.*, 2013.
- [4] J. A. Santos, A. Carrasco, y J. Martínez, «Feature Selection using Typical Testors applied to Estimation of Stellar Parameters.», vol. 8, p. 23, 2004.
- [5] K. Fukunaga, «Introduction to Statistical Pattern Recognition», vol. 2, 2013.
- [6] K. S. Fu, «Syntactic Pattern Recognition», *Acad. Press Inc*, 1974.
- [7] C. G. Looney, «Pattern Recognition using Neural Networks: Theory and Algorithms for Engineers and Scientists», *Oxford Univ.*, p. 458, 1997.
- [8] J. Ruiz-Shulcloper, «Reconocimiento lógico combinatorio de patrones: Teoría y aplicaciones.», p. 387, 2009.
- [9] J. Ruiz-Shulcloper y J. A. Carrasco-Ochoa, «Reconocimiento de patrones: Conceptos y metodología», p. 33, 2013.
- [10] J. A. A. Tamayo, «Optimización bajo incertidumbre aplicada al Sector Eléctrico Colombiano Optimización bajo incertidumbre aplicada al Sector Eléctrico Colombiano», 2014.
- [11] R. Pico-Peña, G. Almagro-Calvizo, y J. Mena-Portales, «Aplicación de Técnicas de Reconocimiento de patrones con enfoque-lógico-combinatorio en la clasificación de un complejo de géneros de hongos patógenos de la caña de azúcar.», p. 6, 1998.
- [12] L. Vega-Alvarado y M. Ortiz-Posadas, «Análisis y clasificación de pacientes con fisuras de paladar mediante un concepto de tipicidad y contraste.», p. 4, 1999.
- [13] S. López-Pérez, M. Lazo-Cortes, y H. . Estrada-García, «Electrodiagnóstico médico

- utilizando las herramientas de reconocimiento de patrones.», pp. 237-244, 1997.
- [14] J. Ruiz-Shulcloper, R. Pico-Peña, y S. R. Morán, «Pronóstico gasopetrolífero en la asociación ofiolítica cubana aplicando modelación matemática.», vol. 33, pp. 447-467, 1994.
- [15] R. Pico-Peña, L. Álvarez, y M. Cotilla, «Zonación sísmica de la isla de Cuba mediante algoritmos de clasificación lógico-combinatorios», pp. 245-250, 1997.
- [16] J. Anil-K, R. P. W. Duin, y J. Mao, «Statistical pattern recognition: A review. Pattern Analysis and Machine Intelligence», vol. 22, pp. 4-37, 2000.
- [17] T. Patvilis, «Structural pattern recognition», p. 385, 2013.
- [18] Y. Reyes-González y N. Martínez-Sanchez, «Algoritmos Conceptuales: Una perspectiva para el modelado del estudiante en los Sistemas Tutoriales Inteligentes.», 2014.
- [19] A. Lias-Rodríguez y A. Pons-Porrata, «Un Nuevo Algoritmo de Escala Exterior para el Cálculo de los Testores Típicos», p. 10, 2009.
- [20] N. Martínez-Sanchez, M. García-Lorenzo, y Z. García-Valdivia, «El enfoque lógico combinatorio: Una perspectiva para la selección de los rasgos relevantes y la organización de los casos en un Sistema Basado en Casos.», 2011.
- [21] J. Ruiz-Shulcloper y S. Vega-Pons, «Índices de validación y combinación de algoritmos conceptuales», 2012.
- [22] E. Alba-Cabrera, «Nuevas extensiones del concepto de testor para diferentes tipos de funciones de semejanza», *Tesis para obtener el grado Dr. en Ciencias Matemáticas, ICIMAF, Cuba*, 1997.
- [23] M. Lazo-Cortes, J. Ruiz-Shulcloper, y E. Alba-Cabrera, «An overview of the evolution of the concept of testor», vol. 34, pp. 753-762, 2001.
- [24] E. Alba-Cabrera, J. Ruiz-Shulcloper, y M. Lazo-Cortes, «Introducción al Reconocimiento de Patrones (Enfoque Logico-Combinatorio)», *CINVESTAV-IPN*, 1995.
- [25] J. Ruiz-Shulcloper, A. G. Arenas, y F. M. Trinidad, *Enfoque lógico combinatorio al reconocimiento de patrones*. 1999.
- [26] J. Ruiz-Shulcloper, J. A. Carrasco-Ochoa, y José Francisco Martínez-Trinidad, «Clasificadores supervisados basados en precedencias parciales», p. 27, 2015.

- [27] S. Pinto y N. Queipo, «3. Clasificación no supervisada», pp. 1-12, 2001.
- [28] M. O. Rodríguez-Cotilla, «Un recorrido por la sismología de Cuba», 2008.
- [29] A. Bravo, «Algoritmo CT para el cálculo de los testores típicos en una matriz k-valente», vol. IV, p. 22, 1983.
- [30] G. Sánchez-Díaz, «Desarrollo y programación de algoritmos eficientes (secuenciales y paralelos) para el cálculo de los testores típicos de una matriz básica», *Tesis Maest. en Ciencias la Comput.*, 1997.
- [31] J. Ruiz-Shulcloper y L. Águila-Feros, «Algoritmo CC para la elaboración de la información k-valente en problemas de Reconocimiento de Patrones», *Rev. Ciencias Matemáticas*, 1984.
- [32] J. Ruiz-Shulcloper, A. Bravo-Martínez, y L. Águila-Feros, «Algoritmos BT y TB para el cálculo de todos los testores típicos.», vol. VI, n.º 2, 1985.
- [33] E. Alba-Cabrera, J. Ibarra-Fiallo, S. Godoy-Calderon, y F. Cervantes-Alonso, «YYC : A Fast Performance Incremental Algorithm for Finding Typical Testors», pp. 416-423, 2014.
- [34] Y. Santiesteban-Alganza y A. Pons-Porrata, «LEX: Un nuevo algoritmo para el cálculo de los testores típicos.», vol. 21, n.º 1, p. 13, 2003.
- [35] J. Ruiz-Shulcloper y R. Pico-Peña, «Un nuevo enfoque en la construcción de sistemas de reconocimiento: Los Sistemas Herramientas.», vol. 13, 1992.
- [36] A. Fuentes-Rodríguez y J. Ruiz-Shulcloper, «ACLAPIP: Un sistema de programas para la clasificación basado en el peso de los patrones», vol. VII, n.º 2, p. 71, 1986.
- [37] J. Ruiz-Shulcloper, R. Pico-Peña, R. López-País, M. S. Lazo-Cortés, M. B. Boggiano-Castillo, E. Barreto-Fiú, A. Santana-Machado, L. Álvarez-Gómez, y T. Chuy-Rodríguez, «PROGNOSIS y sus aplicaciones a las geociencias.», vol. 13, pp. 124-144, 1993.
- [38] R. Pico-Peña y J. Ruiz-Shulcloper, «SELECTOR: Una herramienta de trabajo para especialistas no matemáticos.», *Rev. CENIC Ciencias Biológicas*, vol. 19, n.º 3, pp. 194-196, 1988.
- [39] V. N. Dobrinin y E. N. Cheremesina, «Métodos matemáticos y dispositivos computacionales en las investigaciones de pronóstico geológico (en ruso)», 1988.
- [40] C. Lorena-Corso, «Aplicación de algoritmos de clasificación supervisada usando

- Weka», p. 11, 2009.
- [41] S. R. Garner, «WEKA: The Waikato Environment for Knowledge Analysis», *Proc. New Zeal. Comput. Sci.*, pp. 57-64, 1995.
- [42] E. A. Yero-Oses, Y. Reyes-González, y N. Martínez-Sánchez, «CEPAR: Un sistema herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones.», p. 18, 2016.
- [43] «NetBeans», 2016. [En línea]. Disponible en: <https://netbeans.org/>.
- [44] «Java», 2016. [En línea]. Disponible en: <https://www.java.com/es/about/>. [Accedido: 05-jun-2016].
- [45] V. Paradigm, «Visual Paradigm Essential», 2016. [En línea]. Disponible en: <https://www.udemy.com/visual-paradigm-essential/>.
- [46] M. Turoff y H. Listone, «The Delphi Method. Techniques and Applications», *Adison Wesley*, p. 640, 1975.
- [47] J. Landeta, «El método Delphi: Una técnica de previsión para la incertidumbre», *Ed. Ariel Pract.*, 1999.
- [48] E.-A. C. Silbeto y J.-C. Del-Toro-Ríos, «Las bases de registro, flujos y momentos del sistema de contabilidad gubernamental en el nivel provincial», n.º 192, 2015.
- [49] A.-R. Hernández-Leonard, «Evaluación de la satisfacción con el servicio de capacitación del INIMET.», vol. 1, pp. 18-27, 2013.
- [50] R. Pico-Peña, «PROGNOSIS. Sistema Herramienta de Reconocimiento de Patrones», Instituto de Matemática, Cibernética y Computación, 1995.
- [51] J. S. Marshall y M. A. Thomson, «The Pandora software development kit for pattern recognition», *Eur. Phys. J. C*, vol. 75, n.º 9, p. 439, sep. 2015.
- [52] I. Talavera-Bustamante, L. Bustio-Martínez, Y. Coma-Peña, y N. Hernández-González, «QUIMIOMETRIX II, una plataforma automatizada para el procesamiento multivariante de datos químicos y bioquímicos. Experiencias de aplicación», *Rev. Cuba. Química*, vol. XXV, n.º 3, 2013.
- [53] R. Rakotomalala, «TANAGRA: a free software for research and academic purposes», vol. 2, pp. 697-702.
- [54] R. P. W. Duin y D. M. J. Tax, «PRTools – A Matlab toolbox for pattern recognition», 2013. .

---

## BIBLIOGRAFÍA

---

1. Guzmán Trampe, J. E. Extensión de un lenguaje formal(LCARS) para especificar problemas de clasificación y de selección de rasgos, mediante la construcción de un intérprete optimizado. 121 (2009).
2. Santos, J. A., Carrasco, A. & Martínez, J. Feature Selection using Typical Testors applied to Estimation of Stellar Parameters. **8**, 23 (2004).
3. Mery, D., Da-Silva, R. R., P-Cobola, L. & Joao M. Rebello. Detección de fallas en piezas fundidas usando una metodología de reconocimiento de patrones. 13 (2003).
4. Fukunaga, K. Introduction to Statistical Pattern Recognition. **2**, (2013).
5. Matich, D. J. Redes Neuronales: Conceptos Básicos y Aplicaciones. 55 (2001).
6. Ruiz-Shulcloper, J. Acerca del surgimiento del Reconocimiento de Patrones en Cuba About the appearance of Pattern Recognition in Cuba. **7**, 169-192 (2013).
7. Ruiz-Shulcloper, J. & Carrasco-Ochoa, J. A. Reconocimiento de patrones: Conceptos y metodología. 33 (2013).
8. Tamayo, J. A. A. Optimización bajo incertidumbre aplicada al Sector Eléctrico Colombiano Optimización bajo incertidumbre aplicada al Sector Eléctrico Colombiano. (2014).
9. Pico-Peña, R., Almagro-Calvizo, G. & Mena-Portales, J. Aplicación de Técnicas de Reconocimiento de patrones con enfoque-lógico-combinatorio en la clasificación de un complejo de géneros de hongos patógenos de la caña de azúcar. 6 (1998).
10. Vega-Alvarado, L. & Ortiz-Posadas, M. Análisis y clasificación de pacientes con fisuras de paladar mediante un concepto de tipicidad y contraste. 4 (1999).
11. López-Pérez, S., Lazo-Cortes, M. & Estrada-García, H. . Electrodiagnóstico médico utilizando las herramientas de reconocimiento de patrones. 237-244 (1997).
12. Ruiz-Shulcloper, J., Pico-Peña, R. & Morán, S. R. Pronóstico gasopetrolífero en la asociación ofiolítica cubana aplicando modelación matemática. **33**, 447-467 (1994).
13. Pico-Peña, R., Álvarez, L. & Cotilla, M. Zonación sísmica de la isla de Cuba mediante algoritmos de clasificación lógico-combinatorios. 245-250 (1997).
14. K-Jain, A. Statical Pattern Recognition: A Review. 105 (1999).
15. Lias-Rodríguez, A. & Pons-Porrata, A. Un Nuevo Algoritmo de Escala Exterior para el Cálculo de los Testores Típicos. 10 (2009).
16. Lazo-Cortes, M., Ruiz-Shulcloper, J. & Alba-Cabrera, E. An overview of the evolution of the concept of testor. **34**, 753-762 (2001).

17. Ruiz-Shulcloper, J., Arenas, A. G. & Trinidad, F. M. *Enfoque lógico combinatorio al reconocimiento de patrones*. (1999).
18. Bravo, A. Algoritmo CT para el cálculo de los testores típicos en una matriz k-valente. **IV**, 22 (1983).
19. Ruiz-Shulcloper, J., Bravo-Martínez, A. & Águila-Feros, L. Algoritmos BT y TB para el cálculo de todos los testores típicos. **VI**, (1985).
20. Yero-Oses, E. A., Reyes-González, Y. & Martínez-Sanchez, N. CEPAR:Un sistema herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones. 18 (2016).
21. García-Morate, D. Manual de Weka. 71 (2008).
22. Lorena-Corso, C. Aplicación de algoritmos de clasificación supervisada usando Weka. 11 (2009).
23. Ruiz-Shulcloper, J. Reconocimiento lógico combinatorio de patrones: Teoría y aplicaciones. 387 (2009).
24. Fuentes-Rodríguez, A. & Ruiz-Shulcloper, J. ACLAPIP : Un sistema de programas para la clasificación basado en el peso de los patrones. **VII**, 71 (1986).
25. Pico-Peña, R. PROGNOSIS:Sistema Herramienta de Reconocimiento de Patrones. 80 (1995).
26. Ruiz-Shulcloper, J., Carrasco-Ochoa, J. A. & José Francisco Martínez-Trinidad. Clasificadores supervisados basados en precedencias parciales. 27 (2015).
27. Pinto, S. & Queipo, N. 3. Clasificación no supervisada. 1-12 (2001).
28. Alba-Cabrera, E., Ibarra-Fiallo, J., Godoy-Calderon, S. & Cervantes-Alonso, F. YYC : A Fast Performance Incremental Algorithm for Finding Typical Testors. 416-423 (2014).
29. Santiesteban-Alganza, Y. & Pons-Porrata, A. LEX : Un nuevo algoritmo para el cálculo de los testores típicos. **21**, 13 (2003).
30. Martínez-Sanchez, N., García-Lorenzo, M. & García-Valdivia, Z. El enfoque lógico combinatorio: Una perspectiva para la selección de los rasgos relevantes y la organización de los casos en un Sistema Basado en Casos. (2011).
31. Reyes-González, Y. & Martínez-Sanchez, N. Algoritmos Conceptuales : Una perspectiva para el modelado del estudiante en los Sistemas Tutoriales Inteligentes. (2014).
32. Ruiz-Shulcloper, J. & Vega-Pons, S. Índices de validación y combinación de algoritmos conceptuales. (2012).
33. Paradigm, V. Visual Paradigm Essential. (2016). en <<https://www.udemy.com/visual-paradigm-essential/>>

34. Valdés-Medina, L. & Jaime, L. R.-M. Componente para la extracción de registros de eventos en formato XES del Sistema de Gestión de Ensayos Clínicos SIGEC. 68 (2015).
35. Ruiz-Shulcloper, J. & Pico-Peña, R. Un nuevo enfoque en la construcción de sistemas de reconocimiento: Los Sistemas Herramientas. **13**, (1992).
36. Ruiz-Shulcloper, J. *et al.* PROGNOSIS y sus aplicaciones a las geociencias. **13**, 124-144 (1993).
37. NetBeans. (2016). en <<https://netbeans.org/>>
38. Dobrinin, V. N. & Cheremesina, E. N. Métodos matemáticos y dispositivos computacionales en las investigaciones de pronóstico geológico (en ruso). (1988).
39. Pico-Peña, R. & Ruiz-Shulcloper, J. SELECTOR : Una herramienta de trabajo para especialistas no matemáticos. *Rev. CENIC Ciencias Biológicas* **19**, 194-196 (1988).
40. Pico-Peña, R. PROGNOSIS. Sistema Herramienta de Reconocimiento de Patrones. (1995).
41. Guzmán-Trampe, J. E. Extensión de un lenguaje formal (LCARS) para especificar problemas de clasificación y de selección de rasgos, mediante la construcción de un intérprete optimizado. (2009).
42. Garner, S. R. WEKA: The Waikato Environment for Knowledge Analysis. *Proc. New Zeal. Comput. Sci.* 57-64 (1995).
43. Marshall, J. S. & Thomson, M. A. The Pandora software development kit for pattern recognition. *Eur. Phys. J. C* **75**, 439 (2015).
44. Talavera-Bustamante, I., Bustio-Martínez, L., Coma-Peña, Y. & Hernández-González, N. QUIMIOMETRIX II, una plataforma automatizada para el procesamiento multivariante de datos químicos y bioquímicos. Experiencias de aplicación. *Rev. Cuba. Química* **XXV**, (2013).
45. Rakotomalala, R. TANAGRA: a free software for research and academic purposes. **2**, 697-702
46. Duin, R. P. W. & Tax, D. M. J. PRTools – A Matlab toolbox for pattern recognition. (2013). doi:[www.prtools.org](http://www.prtools.org)
47. Ruiz-Shulcloper, J., García-Borroto, M., Carrasco-Ochoa, J. A., José Francisco Martínez-Trinidad & M Medina-Perez. An Efficient Algorithm for Mining Discriminative Regularities and its Application in Supervised Classification. *Pattern Recognition*. (2010).
48. Carvajal-Hernández, JJ Sánchez-Fernández, L., Villa-Vargas, L., Carrasco-Ochoa, J. A. & José Francisco Martínez-Trinidad. Water quality assessment in shrimp culture using an analytical hierarchical process. *Ecol. Indic.* (2013).
49. Shawe-Taylor, J. & Cristianini, N. Kernel Methods for Pattern Analysis. (2004).

50. Fu, K. S. Syntactic Pattern Recognition. *Acad. Press Inc* (1974).
51. Looney, C. G. Pattern Recognition using Neural Networks: Theory and Algorithms for Engineers and Scientists. *Oxford Univ.* 458 (1997).
52. Anil-K, J., Duin, R. P. W. & Mao, J. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence.* **22**, 4-37 (2000).
53. Patvilis, T. Structural pattern recognition. 385 (2013).
54. Alba-Cabrera, E. Nuevas extensiones del concepto de testor para diferentes tipos de funciones de semejanza. *Tesis para obtener el grado Dr. en Ciencias Matemáticas, ICIMAF, Cuba* (1997).
55. Alba-Cabrera, E., Ruiz-Shulcloper, J. & Lazo-Cortes, M. Introduccion al Reconocimiento de Patrones (Enfoque Logico-Combinatorio). *CINVESTAV-IPN* (1995).
56. Rodríguez-Cotilla, M. O. Un recorrido por la sismología de Cuba. (2008).
57. Sánchez-Díaz, G. Desarrollo y programación de algoritmos eficientes (secuenciales y paralelos) para el cálculo de los testores típicos de una matriz básica. *Tesis Maest. en Ciencias la Comput.* (1997).
58. Ruiz-Shulcloper, J. & Águila-Feros, L. Algoritmo CC para la elaboración de la información k-valente en problemas de Reconocimiento de Patrones. *Rev. Ciencias Matemáticas* (1984).
59. Java. (2016). en <<https://www.java.com/es/about/>>
60. Landeta, J. El método Delphi: Una técnica de previsión para la incertidumbre. *Ed. Ariel Pract.* (1999).
61. Turoff, M. & Listone, H. The Delphi Method. Techniques and Applications. *Adison Wesley* 640 (1975).
62. Hernández-Leonard, A.-R. Evaluación de la satisfacción con el servicio de capacitación del INIMET. **1**, 18-27 (2013).
63. Silbeto, E.-A. C. & Del-Toro-Ríos, J.-C. Las bases de registro, flujos y momentos del sistema de contabilidad gubernamental en el nivel provincial. (2015).

**Anexo 1 Encuesta realizada a los especialistas**

**Estimado compañero/a:**

Nos dirigimos a usted con el objetivo de desarrollar una encuesta que forma parte de la aplicación del Método de Valoración por Especialistas, que se estará desarrollando en el trabajo de diploma: Algoritmos de escala interior para el cálculo de los testores típicos como extensión de la herramienta CEPAR. En la cual se medirá con su ayuda y profesionalidad el correcto desarrollo y funcionamiento de la propuesta. De antemano se le asegura que nadie podrá saber quién es el encuestado. Valore el grado de factibilidad de cada pregunta o afirmación de acuerdo a la siguiente escala: Muy Adecuado (A1); Bastante Adecuado (A2); Adecuado (A3); Poco Adecuado (A4) y No adecuado (A5).

**Gracias de antemano. Atentamente,**

**Leonel González**

Preguntas	Criterios del especialista				
	A1	A2	A3	A4	A5
¿Considera adecuado plantear que: se dificulta la comparación para la selección de un algoritmo para el cálculo de los Testores Típicos dado la inexistencia de una plataforma que permita realizar estas comparaciones?					
¿Considera apropiada la incorporación de variados algoritmos para el Cálculo de todos los Testores Típicos en una herramienta?					
¿Considera adecuado en el RLCP el peso informacional de los rasgos a partir de todos los Testores Típicos?					
¿Considera de beneficio para Investigadores del RLCP, y otras áreas a fin, la propuesta planteada en la presente investigación?					
¿Considera correcta la selección de la herramienta CEPAR para incorporar los algoritmos para el cálculo de todos los Testores					

Típicos?					
----------	--	--	--	--	--

## **Anexo 2 Encuesta realizada a estudiantes y profesores del reconocimiento de patrones.**

### ***Estimados/as:***

Estamos realizando una encuesta con el objetivo de ayudar en el desarrollo y validación de la propuesta de solución del trabajo de diploma: Algoritmos de escala interior para el cálculo de los testores típicos como extensión de la herramienta CEPAR. De antemano se le asegura que nadie conocerá quién es el encuestado. Seguros de su interés y esperanzados de poder contar con su disposición, le pedimos que respondan las siguientes preguntas.

Gracias de antemano.

Atentamente,

Leonel González

1. Marque con una X, atendiendo a si usted es:

Estudiante

Profesor

Especialista

2. Conoce que es el Reconocimiento de Patrones:

Si

No

3. Conoce que es el Reconocimiento Lógico Combinatorio de Patrones (RLCP).

Si

No

4. Conoce herramientas informáticas para resolver problemas el RLCP

Si

No

5. Si la anterior es sí. Mencione las que conoce para el RLCP

6. Conoce la herramienta CEPAR.

Si

No

He trabajado con ella

7. Considera apropiado incluir en una herramienta varios algoritmos de escala interior para el cálculo de todos los Testores Típicos (TT)

Si

No

Me da lo mismo

8. ¿Qué algoritmos de escala interior para el cálculo de todos los TT no deben faltar, según su criterio?

9. ¿Satisface sus necesidades como (estudiante | investigador | docente) una herramienta con varios algoritmos para el cálculo de todos los Testores Típicos?

Me satisface mucho

No me satisface tanto

Me da lo mismo

Me disgusta más de lo que me satisface

No me satisface

No sé qué decir

10. ¿Si usted fuera a realizar un nuevo proyecto preferiría no utilizar la herramienta propuesta para llevar a cabo el cálculo de todos los Testores Típicos?

Si

No

No sé

11. ¿ Si usted necesitara desarrollar una aplicación en la que se lleve a cabo el cálculo de todos los Testores Típicos usaría esta propuesta?

Si

No

No sé

12. ¿Qué es lo que más te gusta de la propuesta?

13. ¿Qué es lo que más te disgusta de la propuesta?

### **Resultados de interés obtenidos de la aplicación de la encuesta**

Partiendo de la encuesta los algoritmos para el cálculo de todos los testores típicos según el criterio de los usuarios, que no debieran dejar de incluirse en la propuesta son:

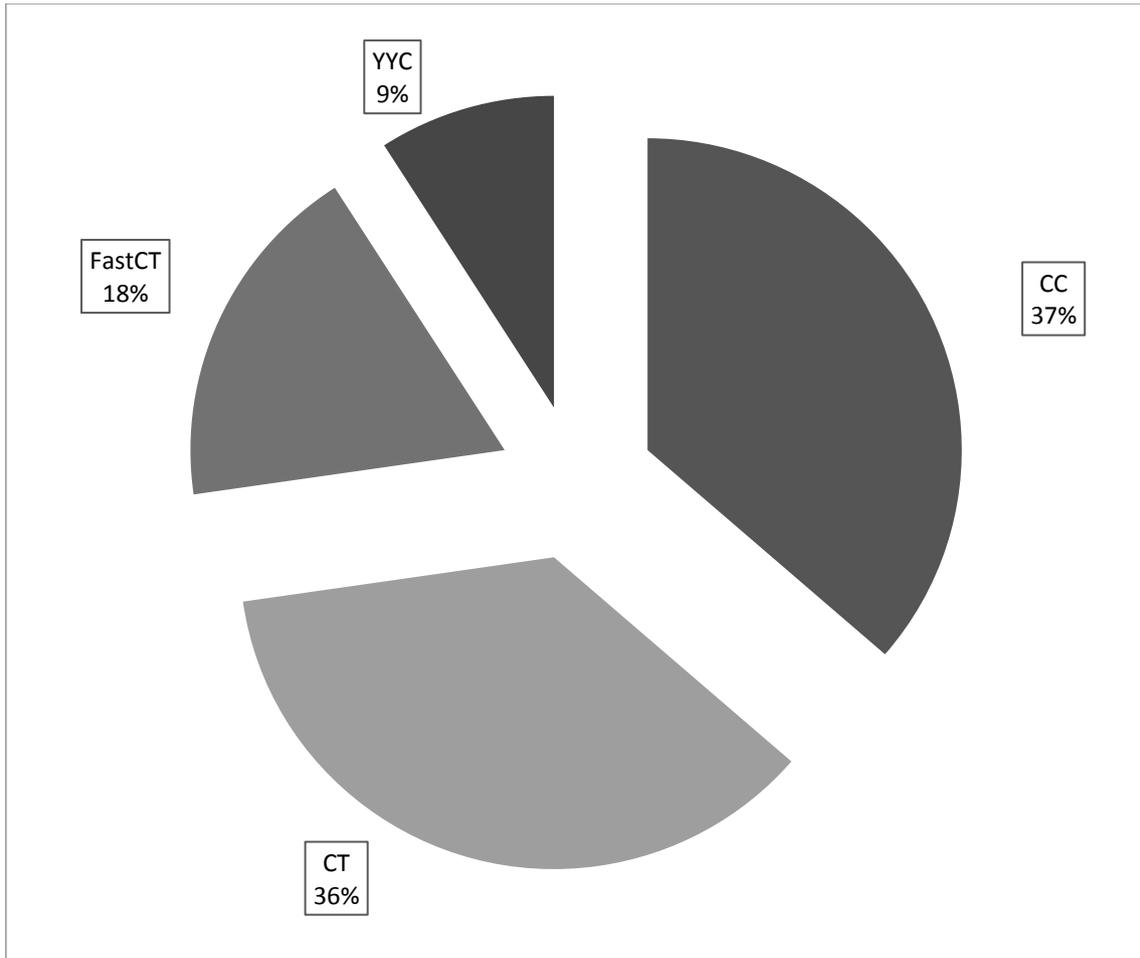


Figura 18 Porcentaje de los algoritmos.

### Anexo 3 Análisis de las herramientas existentes para el RLCP.

La tabla presenta un resumen de las principales características de algunos de los sistemas empleados en la solución de problemas del Reconocimiento de Patrones

	1	2	3	4	5
ACLAPIP [36]	Si	No	Si	No	No
ALISA[39]	No	No	No	Si	No
SELECTOR[38]	Si	No	Si	No	No
PROGNOSIS[50]	Si	No	Si	No	No
LCARS[1]	Si	No	Si	No	No
WEKA[41]	No	Si	Si	No	Si
PANDORA[51]	No	Si	-	-	Si
QUIMIOMETRIX[52]	No	Si	No	Si	-

TANAGRA[53]	No	Si	-	-	Si
PRTtools[54]	No	Si	No	Si	Si
CEPAR[42]	Si	Si	Si	No	Si

**Características analizadas:**

1. Desarrolladas en base a las teorías del Reconocimiento Lógico Combinatorio de Patrones
2. Permite la incorporación de nuevas funcionalidades, algoritmos, tipos de rasgos y dominios de definición de las variables.
3. Permite el trabajo simultáneo con variables cualitativas y cuantitativas
4. Presupone un tipo especial de espacio de representación para los datos (métrico, vectorial, euclidiano, booleano, etc.).
5. Soporte multiplataforma (Sistema Operativos).