



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD 2**

*Trabajo de Diploma para optar por el título de Ingeniero en  
Ciencias Informáticas*

***Extensión de la herramienta CEPAR con algoritmos de  
escala exterior para el cálculo de Testores Típicos***

**Autoras:**

- ~ Eyidey Pérez Hernández
- ~ Paloma García Figarola

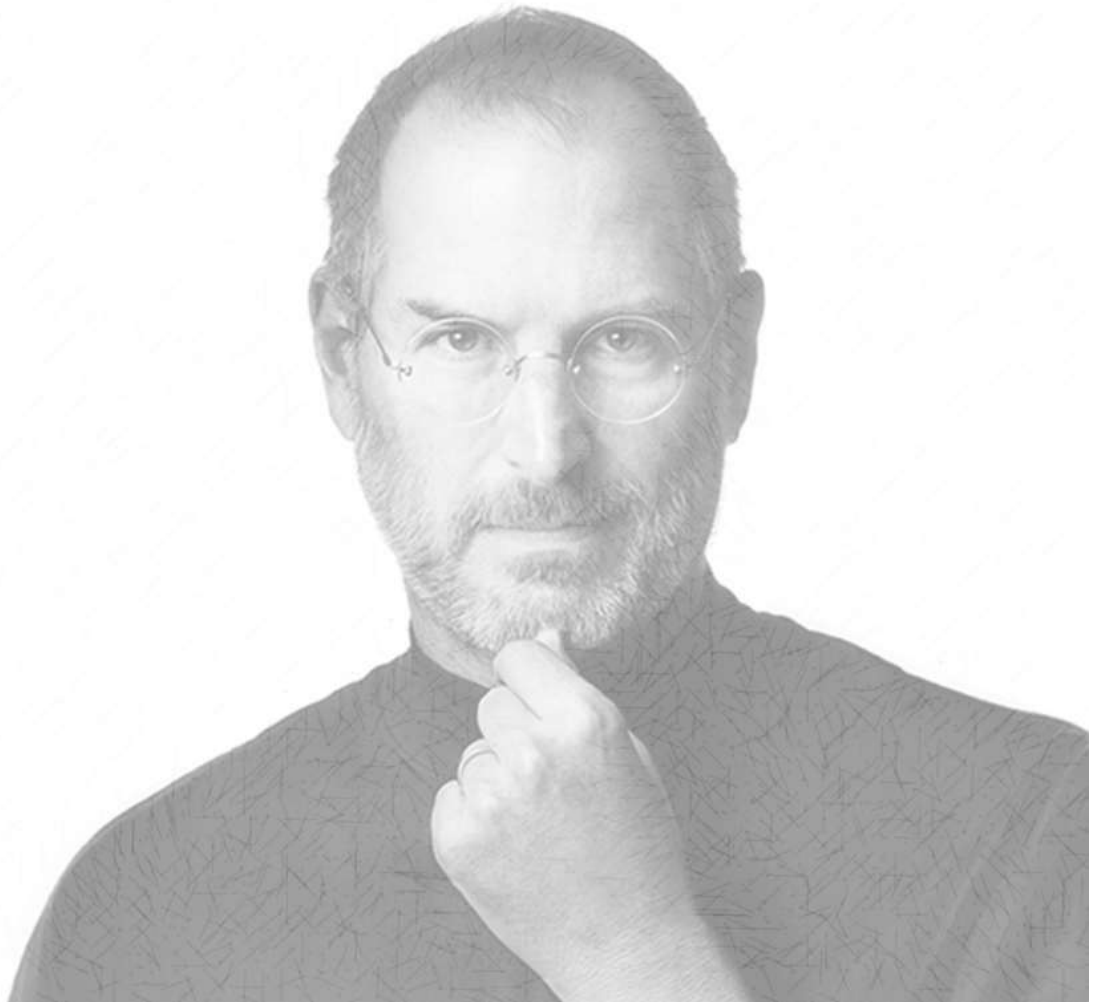
**Tutores:**

- ~ Ing. Ernesto Alejandro Yero Oses
- ~ Lic. Lester Rodríguez Vallejo

**La Habana, junio de 2016**

**“Año 58 de la Revolución”**

*“... conectados al futuro, conectados a la Revolución.”*



*“Ten el coraje de seguir tu corazón y tu intuición, ellos saben de alguna manera lo que realmente quieres ser, todo lo demás es secundario. ”*

*Steve Jobs.*

---

## Declaración de Autoría

---

Declaramos ser autoras 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 a los \_\_\_\_ días del mes de julio del año 2016.

---

Eyidey Pérez Hernández

Firma del Autor

---

Paloma García Figarola

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 (UCI) en el año 2015. Pertenece al Centro de Informática Médica (CESIM). Trabaja la línea de investigación 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 de Técnicas de Programación de la Universidad de las Ciencias Informáticas. Trabaja la línea de investigación Reconocimiento Lógico Combinatorio de Patrones y Teoría de los Conjuntos Aproximados.

---

## Agradecimientos

---

### **Agradecimientos de Eyidey.**

*Primero que todo agradezco a mis padres por apoyarme en cada decisión que tomé, por saber ser padres ante todo, por estar siempre ahí cuando los necesité porque de no ser por ellos no hubiese llegado tan lejos, por saberme guiar por un buen camino, por enseñarme a respetar, a querer y a nunca darme por vencida, a ellos más que a nada agradezco este logro que también es tuyo.*

*A mi hermano Erick por su cariño y por querer seguir mi ejemplo, eso me dio mucha fuerza...te quiero.*

*A mi novio Leo por tener ese gran corazón y aguantarme todo este tiempo, por su apoyo, por su amor, porque cada vez que lo necesité estaba ahí para mí sin importarle nada y sobre todo por hacerme tan feliz...te amo.*

*En general a toda mi familia por creer en mí y darme su apoyo.*

*A la gente del grupo en especial a Víctor que sin su ayuda no lo hubiese logrado.*

*Agradezco también a todos mis amigos Ismaray, Dayana, Daniel por su cariño, sus consejos, por todos los malos y buenos momentos que hemos pasado juntos.*

*A mi compañera de tesis Paloma porque solo ella sabe el trabajo que pasamos para estar aquí el día de hoy.*

*A mi tutor Yero por su paciencia, por toda la ayuda que brindó en la realización de este trabajo, por todos los consejos, sugerencias y recomendaciones que fueron de gran ayuda. Muchísimas gracias, estaré eternamente agradecida por todo el apoyo que me ha dado.*

---

## Agradecimientos

---

### **Agradecimientos de Paloma**

*Le doy gracias a Dios por darme fuerzas cuando más las necesité.*

*A mi mamá Ivón, porque siempre creyó en mí y me sacó adelante dándome su mejor ejemplo. Por estar siempre a mi lado, por sus consejos, sus regaños y enseñarme que no importa las veces que uno caiga, lo que importa es levantarse.*

*A mi abuela querida Olga por su entrega y dedicación a la familia, por ser el miembro más importante de esta, por sus consejos y sus peleas. Por su amor incondicional.*

*A mi tío Albert, por ser más que mi tío, por ser mi padre, por tener un corazón puro y lleno de amor a pesar de la distancia, gracias por apoyarme en todo.*

*A mi abuelo Gil, por ser el mejor hombre que conocí, por darme su amor hasta donde pudo.*

*A mis tíos Flor y William porque sin ellos parte de esta meta no sería posible.*

*A mis tíos Cary y Napoleón por sus palabras de aliento, por su confianza.*

*A mi novio Giovanni, por su confianza, paciencia y sobre todo su amor.*

*A mis primeras amigas de la UCI, Yani y Mayde, por enseñarme el valor de la amistad, por los buenos momentos y los malos.*

*A mi super amigo Sosa, por sus consejos, sus regaños, y su eterna preocupación, por siempre darme ánimos y decirme si otros pudieron porque nosotros no.*

*A mis amiguis que siempre me han escuchado y aconsejado, con las que compartí buenos y malos momentos, Lea, Nela, Ada, Lismey, Mirna, Lía y Angela.*

*A los amigos que he encontrado a lo largo de la carrera, José Manuel, Osly, Ledian, Eddy, Glauver, Pachi, Ilisen, Samy, Lionys, Andy, Ramón.*

*A Victor por su mega ayuda, sin él no estaríamos hoy aquí.*

*A mi compañera de tesis Eyidey, por haber sido una excelente compañera de tesis y ayudarme en los momentos de desesperación.*

*A nuestro tutor Yero, por su ayuda y paciencia, y sobre todo por su entrega.*

*A todos, gracias.*

---

## Dedicatoria

---

### ***Dedicatoria de Eyidey***

*Esta tesis va dedicada en especial a mis padres, mi hermano, mi novio, mi familia y a todos los que pensaron que no lo lograría.*

### ***Dedicatoria de Paloma***

*A mi mamá, mi abuela, mi abuelo y mis tíos.*

*A todas mis amistades que estuvieron a mi lado y creyeron en mí. A todos los que de una forma u otra me ayudaron.*

Entre los enfoques más conocidos del Reconocimiento de Patrones se encuentra el Lógico Combinatorio el cual permite trabajar con variables cuantitativas, cualitativas e incluso con valores ausentes. En este enfoque el problema de selección de rasgos y/o objetos se resuelve apoyándose en el peso informacional de los rasgos, comúnmente usando la Teoría de Testores (Testores Típicos), tiene como objetivo encontrar el subconjunto de rasgos que mejor describe los objetos del dominio y que proveen información relevante en el proceso de clasificación.

En la actualidad existen varios algoritmos para el cálculo de todos los Testores Típicos, estos se encuentran dispersos en diferentes herramientas y en ocasiones el enfoque usado por la herramienta para la resolución de los problemas no es el Reconocimiento Lógico Combinatorio de Patrones. Por lo que se dificulta la selección del mejor algoritmo para la solución de un problema dado, al ser engorroso el proceso de comparar los algoritmos.

Se propone la incorporación de los algoritmos para el cálculo de todos los Testores Típicos en la herramienta CEPAR, como extensión de la misma, de esta manera se facilita 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 permite su reutilización en problemas prácticos del Reconocimiento Lógico Combinatorio de Patrones.

**Palabras clave:** reconocimiento de patrones, testores típicos, selección de rasgos y/o objetos, algoritmos de escala exterior.



---

## índice

---

Introducción.....	13
Capítulo 1 . Conceptos del Reconocimiento de Patrones y Sistema Herramienta asociadas .....	19
1.1 Reconocimiento de Patrones .....	19
1.1.1 Conceptos básicos asociados al Reconocimiento de Patrones .....	20
1.1.2 Problemas del Reconocimiento de Patrones .....	22
1.2 Selección de rasgos y/o objetos.....	23
1.3 Teoría de Testores.....	24
1.3.1 Peso informacional de los rasgos en función del concepto de Testor .....	24
1.4 Algoritmos para el cálculo de todos los Testores Típicos .....	26
1.4.1 Algoritmos de Escala Exterior.....	26
1.5 Herramientas para la solución de problemas del Reconocimiento de Patrones .....	30
1.5.1 ALISA.....	31
1.5.2 SELECTOR.....	31
1.5.3 ACLAPIP .....	32
1.5.4 PROGNOSIS.....	33
1.5.5 WEKA.....	34
1.5.6 CEPAR.....	34
1.5.7 Justificación de la selección .....	36
1.6 Herramientas y tecnologías usadas .....	36
1.6.1 Lenguaje de programación JAVA .....	36
1.6.1 Entorno de Desarrollo Integrado NetBeans .....	36
1.6.2 Herramienta CASE Visual Paradigm .....	37
Capítulo 2 . Propuesta de solución para la extensión de la herramienta CEPAR.....	38
2.1 Descripción e Implementación .....	38
2.1.1 Algoritmo BT.....	38
2.1.2 Algoritmo TB.....	39
2.1.3 Algoritmo LEX .....	40

2.1.4 Algoritmo BR .....	42
2.2 Integración de los algoritmos en la herramienta .....	45
Capítulo 3 . Análisis y validación de los resultados.....	51
3.1 Proceso de validación de la solución .....	51
3.2 Validación por expertos.....	51
3.2.1 Proceso de selección de los expertos.....	52
3.2.2 Elaboración del cuestionario.....	56
3.2.3 Análisis y procesamiento de los resultados .....	56
3.3 Resultado experimental a partir del desarrollo de casos de estudio .....	58
3.4 Validación de satisfacción de potenciales usuarios.....	59
3.4.1 Análisis de los resultados .....	61
3.5 Resultados de las pruebas y validaciones .....	62
Conclusiones.....	63
Recomendaciones.....	64
Referencias .....	65
Bibliografía .....	69
Anexos .....	73
Anexo 1 .....	73
Anexo 2 .....	73
Anexo 3 .....	74

---

## Índice de Ecuaciones

---

Ecuación 1. Peso informacional del rasgo.....	25
Ecuación 2. Peso del rasgo dependiendo de la longitud del Testor.....	25
Ecuación 3. Fórmula del Coeficiente de Kendall.....	56
Ecuación 4. Índice de Satisfacción Grupal.....	61

---

## Índice de Figuras

---

Figura 1. Enfoques del RP .....	20
Figura 2. Diagrama de flujo algoritmos de escala exterior .....	27
Figura 3. Paquetes de CEPAR .....	46
Figura 4. Paquetes de CEPAR con algoritmos incorporados.....	46
Figura 5. Diagrama de clases Algoritmo BT .....	47
Figura 6. Diagrama de clases Algoritmo TB .....	47
Figura 7. Diagrama de clases Algoritmo LEX .....	48
Figura 8. Diagrama de clases algoritmo BR .....	48
Figura 9. Selección de rasgos .....	49
Figura 10. Selección del algoritmo a emplear.....	49
Figura 11. Resultado del algoritmo aplicado.....	50
Figura 12. Procesos de validación.....	51
Figura 13. Coeficiente de Competencia.....	55
Figura 14. Gráfica de adecuación de preguntas .....	58
Figura 15. Porcentaje del nivel de adecuación .....	58
Figura 16. Porcentaje de algoritmos .....	73

---

## Índice de Tablas

---

Tabla 1. Tiempo del funcionamiento de los algoritmos para varias matrices básicas .....	30
Tabla 2. Tabla para el procesamiento del grado de conocimiento sobre el tema.....	53
Tabla 3. Procesamiento de los aspectos que pueden influir en el nivel de argumentación .....	54
Tabla 4. Tabla patrón para la determinación del nivel de argumentación .....	54
Tabla 5. Resultados del procesamiento para la determinación del coeficiente de competencia de los expertos.....	55
Tabla 6. Caracterización de los expertos que participaron en la consulta.....	55
Tabla 7. Resultado de las encuestas a expertos .....	56
Tabla 8. Resultados de ejecutar los distintos algoritmos sobre un mismo juego de datos .....	59
Tabla 9. Cuadro Lógico de ladov modificado por los autores .....	60
Tabla 10. Escala de satisfacción .....	60
Tabla 11. Intervalos de satisfacción.....	61
Tabla 12. Características de los encuestados de ladov.....	61

Cuando el ser humano entra en contacto por primera vez con un objeto, su mente percibe instantáneamente a este como un conjunto de datos sensoriales formados por los atributos propios del objeto tales como: color, forma, tamaño, olor, textura, sonido, etc. Esa percepción constituye el primer paso en el proceso para reconocer dicho objeto. Una vez que un objeto es conocido, el cerebro tiene la capacidad de reconocerlo en ocasiones posteriores. El proceso de reconocimiento ocurre cuando, ante la presencia de un nuevo elemento, el conjunto de atributos que se perciben en él, resulta notoriamente semejante al conjunto de atributos del elemento previamente conocido. Ante esas condiciones la mente asocia al nuevo elemento con él o los anteriormente conocidos y es entonces cuando se le reconoce. [1]

Se puede decir que reconocer patrones es una actividad de carácter involuntario en muchos seres vivos, pero no es hasta finales de los años cincuenta que se comienza a conformar el Reconocimiento de Patrones (RP) como una ciencia. Específicamente según la bibliografía consultada [2], con la publicación del libro *Principios de la Neurodinámica de Rosenblatt* en 1965.

En sus inicios el RP estuvo fuertemente vinculado a la identificación, al reconocimiento y la clasificación de imágenes. No es por ello extraño que, en muchas ocasiones, se identifique esta disciplina con todo lo relacionado exclusivamente al procesamiento y análisis de imágenes. [3]

En la actualidad no existe un consenso sobre la definición RP, da muestra de ello las distintas definiciones dadas por diversos autores [2] [4] [5], lo que ha ocasionado que sus problemas sean tratados desde diferentes perspectivas. Dando origen a lo que autores denominan enfoques del RP, siendo estos tres los más conocidos [6] [7] [8]:

- Enfoque Estadístico.
- Enfoque Sintáctico Estructural.
- Enfoque Lógico Combinatorio.

En los últimos años se ha incrementado el interés por esta ciencia, sobre todo por la demanda de aplicaciones computacionales en diversas áreas relacionadas con: la clasificación de documentos, los pronósticos financieros, la organización y recuperación en bases de datos multimedia, las herramientas para la toma de decisiones, entre otras. [9]

Se entienden por problemas del RP todos aquellos relacionados con la clasificación de objetos y fenómenos, y con la determinación de los factores que inciden en los mismos. Encontrándose así según la bibliografía consultada cuatro familias fundamentales de problemas [2] [6] [7]:

- selección de rasgos y/o objetos;

- clasificación supervisada;
- clasificación no supervisada;
- clasificación parcialmente supervisada;

La selección de rasgos y/o objetos, en muchas ocasiones se considera un paso previo en la modelación de otros problemas, consiste en encontrar los rasgos que proveen información relevante en el proceso de clasificación. En el Reconocimiento Lógico Combinatorio de Patrones (RLCP) esta familia de problemas se resuelve comúnmente usando la Teoría de Testores (o Testores Típicos). [10] [11]

La selección de rasgos y/o objetos ha sido empleada para resolver diferentes problemas, por ejemplo: en la determinación de la influencia de ciertas características socioeconómicas en el análisis de la delincuencia juvenil [12], en la valoración de pacientes pediátricos que presenten malformaciones congénitas de labio y/o paladar hendido [13] [14] [15], en la búsqueda de hongos patógenos de la caña de azúcar [16], en el pronóstico gasopetrolífero en la asociación ofiolítica cubana [17], en la identificación de factores de riesgo asociados con TRALI (daño pulmonar agudo por transfusión) [18], para determinar la zonación sísmica de la isla de Cuba [19], para la clasificación de contenido web y procesamiento de textos [20], en la clasificación de las muestras celulares tomadas del cuello del útero en cancerosas o no [21], así como selección de rasgos para clasificar emociones [22] entre otros.

Es común en casi todos los centros de investigación y universidades con carreras relacionadas a esta disciplina, encontrar grupos de investigadores que trabajen sobre algún enfoque en específico y que desarrollen sus propios modelos y algoritmos para tratar los problemas concretos de los quehaceres cotidianos. [1]

En Cuba el RP tiene sus inicios a finales de los años setenta con dos vertientes fundamentales de trabajos: 1) el procesamiento digital de señales de audio, de voz principalmente, en la Universidad Central de las Villas, en el Ministerio del Interior y en el Instituto de Cibernética, Matemática y Física (ICIMAF); 2) las aplicaciones de herramientas matemáticas a la solución de problemas en las Geociencias y las Ciencias Sociales en el Instituto de Geofísica y Astronomía, en la Universidad de la Habana y en el ICIMAF. [23]

Una década después en el ICIMAF existió 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. El primer grupo trabajó en la extracción de parámetros (patrones en el dominio del tiempo y la frecuencia) de la señal de voz para su reconocimiento. [23]

El segundo grupo de trabajo creó una nueva rama de investigaciones en el RP conocida posteriormente a nivel internacional como “Reconocimiento Lógico Combinatorio de Patrones”. Inicialmente este grupo trabajó en el desarrollo de la Teoría de Testores, una rama de la Lógica Matemática surgida en 1959 en la desaparecida Unión Soviética. La Teoría de Testores se empezó a aplicar en el RP a partir del año 1965 para la selección de rasgos. Este grupo del ICIMAF introdujo una serie de conceptos y algoritmos para la estimación del peso informacional de los rasgos, lo que permitió enfrentar proyectos como la determinación de las causas de la delincuencia juvenil. [23]

En la actualidad existen varios algoritmos para el cálculo de todos los Testores Típicos (TTs), al referirse a los TTs, en la presente investigación quedarán restringidos a los TTs de Zhuravliov [24]. Estos algoritmos se clasifican de acuerdo a su estrategia computacional en dos categorías: algoritmos de escala exterior y de escala interior. Los TTs han sido ampliamente utilizados para evaluar la importancia informacional de los rasgos [25] y como conjunto de apoyo en los algoritmos de clasificación [24]. En Minería de Textos se han usado además para la categorización de textos [24] y construcción de resúmenes de documentos [26].

El RLCP no cuenta con una plataforma (sistema, biblioteca o framework) que contenga implementados varios algoritmos para el cálculo de todos los TTs, estos se encuentran dispersos en distintas herramientas, por lo que se presentan los problemas listados a continuación:

- Cada herramienta establece su propio mecanismo de entrada de información.
- El enfoque usado por la herramienta para la resolución de los problemas no siempre es el RLCP.
- No es posible reproducir, con total veracidad, investigaciones que realicen comparaciones (en cuanto a: tiempos de ejecución) entre diversos algoritmos.

Todo esto provoca que los especialistas tengan que ajustar su modelo a cada herramienta, con el objetivo de poderla emplear en la solución de su problema, ocasionando variaciones en el modelo original. La solución de un mismo problema bajo distintos modelos ocasiona que los resultados no puedan ser comparados con total exactitud. La elaboración de distintos formatos de entrada de una misma información, representa un retraso en las investigaciones.

Muchas de las investigaciones en las que se realizan comparaciones entre diversos algoritmos con distintos juegos de datos, resultan poco reproducibles al no existir una plataforma en la que estos experimentos se puedan comprobar.

No resulta un secreto que el tiempo de respuesta ante determinadas operaciones varía en función del lenguaje de programación empleado, ocasionando diferentes resultados en tiempo de ejecución para un mismo algoritmo sobre un mismo juego de datos implementado en distintos lenguajes.



Considerando todo lo expresado anteriormente se propone como **problema a resolver**: la dispersión de los algoritmos para el cálculo de todos los Testores Típicos dificulta la selección de un algoritmo para ser empleado en la determinación del peso informacional de los rasgos.

Definiéndose como **objeto de estudio**: la selección de rasgos y/o objetos en el Reconocimiento Lógico Combinatorio de Patrones; centrándose en el **campo de acción**: el cálculo de todos los Testores Típicos con algoritmos de escala exterior, para la determinación del peso informacional de los rasgos.

Teniendo como **objetivo general**: implementar, como extensión de una herramienta para la solución de problemas de selección de rasgos y/o objetos, el cálculo de todos los Testores Típicos con algoritmos de escala exterior, facilitando el trabajo con estas técnicas del Reconocimiento Lógico Combinatorio de Patrones.

Para lograr el cumplimiento de este objetivo se proponen como **tareas de investigación**:

- Caracterizar el marco teórico-conceptual de los algoritmos de escala exterior para el cálculo de todos los TTs, así como los sistemas informáticos para la solución de problemas del RP haciendo énfasis en aquellos que permiten su extensión con nuevos algoritmos.
- Realizar la selección de los algoritmos para el cálculo de todos los TTs, destacando los más empleados en investigaciones prácticas.
- Identificar la herramienta (o sistema herramienta), que garantice una mayor reutilización de estos algoritmos, para su extensión con algoritmos de escala exterior para el cálculo de todos los TTs.
- Implementar los algoritmos de escala exterior para el cálculo de todos los TTs para problemas de selección de rasgos y/o objetos, en la herramienta identificada.
- Validar la propuesta de solución mediante el método Delphi de consulta a expertos.
- Validar los resultados de la solución implementada mediante experimentos, a partir de un caso de estudio para evaluar el conjunto de TTs generados por estos algoritmos.
- Validar el nivel de satisfacción en usuarios potenciales de la propuesta desarrollada mediante el método IADOV.

Entre los **beneficios esperados** con el desarrollo de la presente investigación, se encuentran:

- Facilitará la comparación entre los resultados de ejecutar varios algoritmos para el cálculo de todos los TTs, favoreciendo la selección del mejor algoritmo para un problema dado.
- Permitirá la reutilización de los algoritmos implementados en diversos modelos matemáticos atendiendo a las teorías del RLCP según la herramienta seleccionada.

Para el desarrollo del trabajo se utilizan **métodos de investigación**, a continuación, se listan los mismos:

### **Métodos Teóricos:**

Permiten estudiar las características del objeto de investigación que no son observables directamente, facilitan la construcción de modelos e hipótesis de investigación y crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, contribuyendo al desarrollo de las teorías científicas. [27]

**Analítico-Sintético:** se pone de manifiesto en el análisis de documentos y la extracción de los elementos más importantes de los procesos de compilaciones bibliográficas referidas al RP.

**Análisis Histórico-Lógico:** se evidencia en la realización del estudio y análisis de las herramientas y algoritmos existentes en la actualidad para el trabajo con el RLCP.

**Modelación:** se emplea en el diseño del diagrama de flujo de los algoritmos de escala exterior, en la modelación de los diagramas de clases de cada uno de los algoritmos y la integración dentro de la herramienta.

### **Métodos Empírico:**

Describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. [27]

**Entrevista:** se desarrolló un grupo de entrevistas formales no estructuradas para obtener información rápida y precisa de los principales problemas, necesidades y posibles resultados esperados.

**Encuesta:** se realizaron encuestas a un conjunto de usuarios entre ellos estudiantes, expertos, profesores y especialistas para obtener información referente a la propuesta de solución. Estas encuestas se basan en el planteamiento de preguntas escritas y organizadas, donde las repuestas fueron seleccionadas de acuerdo a criterios específicos.

**Medición:** durante la experimentación se utilizó el método Delphi para la valoración de los expertos, a partir de casos de estudio se realizaron pruebas para evaluar el conjunto de TTs generados por estos algoritmos y la técnica Iadov para la satisfacción de los usuarios.

El documento está estructurado en tres capítulos, siendo estos:

### **Capítulo 1. Conceptos del Reconocimiento de Patrones y Sistemas Herramientas**

**asociadas:** Incluye la fundamentación teórica y la descripción de algunos conceptos básicos asociados al RP para su mejor comprensión, así como un análisis de las herramientas del RLCP en problemas de selección de rasgos y además una descripción de los algoritmos de escala exterior para el cálculo de todos los TTs en dichos problemas.

**Capítulo 2. Propuesta de solución para la extensión de la herramienta CEPAR:** se aborda desde el punto de vista teórico cada uno de los algoritmos de escala exterior propuestos a implementar BT, TB, LEX y BR para el cálculo de todos los TTs. Además, se describe el proceso de integración de estos algoritmos en la herramienta seleccionada.

**Capítulo 3. Análisis y validación de los resultados:** se define la validación de la propuesta mediante el método Delphi para conocer el nivel de concordancia entre los expertos, así como pruebas a partir de un caso de estudio para evaluar el conjunto de TTs generados por estos algoritmos y se aplicó la técnica IADOV para saber el grado de satisfacción de los usuarios.

En el presente capítulo, se ofrece una descripción de los aspectos teóricos que son fundamentales para el desarrollo de la investigación. Se estudian conceptos referidos al RP desde la perspectiva de varios autores. Además, se realizan una descripción de los algoritmos de escala exterior para el cálculo de todos los TTs y se analizan algunas de las herramientas para el RLCP que resuelvan problemas de selección de rasgos.

### 1.1 Reconocimiento de Patrones

El RP es una disciplina en el área de la ciencia que pretende reemplazar algunas de las actividades humanas con dispositivos electrónicos. El mismo ha sido definido por varios autores:

**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. [3]

**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. [4]

**Según Fukunaga:** el Reconocimiento de Patrones está relacionado con el proceso de toma de decisiones de un ser humano. El objetivo de este es aclarar los mecanismos complicados de los procesos de toma de decisiones y automatizar las funciones usando computadoras. Sin embargo, debido a la compleja naturaleza del problema, la mayoría de las investigaciones de Reconocimiento de Patrones se han concentrado en los problemas más realistas, como el reconocimiento de caracteres latinos y la clasificación de formas de onda. [5]

En la presente investigación se adopta el concepto de José Ruíz Shulcloper, porque es el autor que muestra una definición más cercana al RLCP.

Como se plantea anteriormente el RP tiene distintos enfoques, estos enfoques se pueden representar atendiendo a la forma o mecanismo empleado para representar los rasgos como muestra la siguiente imagen:



**Figura 1. Enfoques del RP**

El presente trabajo se enmarca en el RLCP. Este enfoque se basa en la idea de que el modelado del problema debe ser lo más cercano posible a la realidad del mismo, sin hacer suposiciones que no estén fundamentadas. Uno de los aspectos esenciales del enfoque es que los atributos utilizados para describir a los objetos de estudio deben ser tratados cuidadosamente para no realizar operaciones que resulten antinaturales respecto al problema que están representando. Este tratamiento cuidadoso permite trabajar con atributos cualitativos y cuantitativos e incluso con ausencia de información. [28]

La Lógica Matemática, la Teoría de Testores, la Teoría Clásica de Conjuntos, la Teoría de los Subconjuntos Difusos, la Teoría Combinatoria, la Matemática Discreta en general, constituyen el basamento teórico-matemático en el que se desarrolla el denominado Reconocimiento Lógico Combinatorio de Patrones. 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. [7]

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

A continuación, se enuncian los principales conceptos relacionados al RP para ayudar a una mejor comprensión de la temática abordada en la presente investigación.

Se denomina **objeto** al término empleado en diferentes disciplinas para referirse a los entes sujetos a estudio. [2]

Se entiende el conjunto de todos los posibles objetos admisibles para los propósitos del estudio en cada caso particular como **universo de objetos**. [2]

Se llama **clase** a un conjunto de objetos que constituyen un tipo, una categoría, en el contexto de los objetos del universo. Siempre será un subconjunto propio del universo de objetos. [2]

**Rasgo** (atributo, variable, característica, propiedad, primitivas de un alfabeto) es un factor a tener en cuenta en el estudio de los objetos dados. Constituyen la vía real para poder estudiar, procesar, analizar a los objetos. Es decir, se tienen que estudiar descripciones de los objetos en términos de un conjunto de rasgos. [2]

**Dominio**; es el conjunto de valores admisibles que tiene asociado un rasgo. [2]

**Patrón** es sinónimo de objeto. 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 de otros que no se sabe nada acerca de dichas propiedades. En esos casos suele usarse el término “objeto” para aquellos de los que se desconocen estas propiedades y se les llama “patrones” a aquellos de los que estas propiedades son conocidas. [3]

Para determinar cuan semejante son dos valores de un mismo rasgo, el RLCP establece lo que se denomina **criterio de comparación** de valores de la variable. Un criterio de comparación entre valores de una variable  $X_j$  a una función  $C_j: M_j \times M_j \rightarrow L_j$ , tal que si  $C_j$  es un criterio de disimilaridad se cumple que  $C_j(X_j(O), X_j(O)) = \min\{y\}$  y si  $C_j$  es un criterio de similaridad se cumple que  $C_j(X_j(O), X_j(O)) = \max\{y\}$ , donde  $L_j$  es un conjunto totalmente ordenado,  $j = 1, \dots, n$ . [29]

En el RLCP el método empleado para determinar la similitud (o semejanza) entre 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. [30]

Sea una matriz de entrenamiento ( $ME$ ) que contiene  $m$  objetos descritos en términos de  $n$  rasgos  $\mathfrak{R} = \{X_1, \dots, X_n\}$  distribuidos en  $r$  clases  $\{C_1, \dots, C_r\}$ . Cada rasgo  $X_i$  toma valores en un conjunto  $D_i, i = 1, \dots, n$ . Dado un criterio de comparación de diferencia  $\Psi_i: D_i \times D_i \rightarrow \{0,1\}$  se asocia a cada  $X_i$  ( $0 =$  similar,  $1 =$  disimilar), para todo posible par de objetos que pertenecen a diferentes clases de  $ME$  se crea una **matriz booleana de diferencia** ( $MD$ ). [24]

Sean  $p$  y  $q$  dos filas de  $MD$ ,  $p$  es subfila de  $q$  si en todas las columnas donde  $p$  tiene un 1,  $q$  también lo tiene. Se llama **fila básica**  $p$  de  $MD$ , si ninguna fila en  $MD$  es subfila de  $p$ . La submatriz de  $MD$  que contiene todas las filas básicas (sin repeticiones) se llama **matriz básica** ( $MB$ ). [24]

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

Como se mencionaba inicialmente los problemas del RP son todos aquellos relacionados con la clasificación de objetos y fenómenos, y con la determinación de los factores que inciden en los mismos. A continuación, se muestra una descripción de cada uno de ellos.

#### **Clasificación supervisada:**

Este es el problema más conocido del RP y con el que más se está relacionado, pues mucha de la actividad humana está vinculada, directa o indirectamente, con procesos de clasificación. Dado un universo de objetos y el conocimiento acerca de la existencia de ciertas clases con características (propiedades) de especial interés y una muestra de objetos que pertenecen a cada una de ellas, el problema consiste en determinar para cada uno de los objetos no clasificados las relaciones de pertenencia de los mismos con cada una de las clases. [2]

#### **Clasificación no supervisada:**

El problema de la clasificación no supervisada (problemas de agrupamiento – clustering, structuralization, en inglés) es una de las etapas más importantes en la mayoría de las ciencias, en particular, las naturales y sociales. La Taxonomía es el primer período de trabajo científico en muchas de las áreas del conocimiento y es una de las herramientas necesarias en muchas de las investigaciones en la actualidad. [2]

El problema principal a resolver es encontrar las relaciones entre los objetos de un universo en términos de sus características (rasgos). Estas relaciones se establecen sobre la base del concepto de analogía (similaridad). Quizás uno de los conceptos más importantes en el RP. [2]

El propósito es juntar (agrupar) los objetos según su analogía (parecido, semejanza, cercanía si se está hablando de un espacio de representación con distancia definida). En este sentido se pueden encontrar dos situaciones diferentes, a saber, el número de grupos (clusters) es conocido previamente o no. [2]

#### **Clasificación parcialmente supervisada:**

Esta es quizás la familia de problemas de RP en la que menos estudio se ha realizado. A pesar de la existencia de muchos problemas reales donde aparecen situaciones de este tipo, la clasificación parcialmente supervisada no ha recibido igual atención que las restantes familias de problemas en RP. [2]

El 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 central es clasificar nuevos objetos en estas circunstancias, en la que no se tienen muestras de todas las clases y en las que incluso pudieran existir clases que se desconocen.[2]

Por ejemplo, se sabe que en una habitación hay un grupo de personas conversando. No se conocen las voces de todas las personas, que se encuentran en la habitación. Se puede incluso no conocer cuántas personas se encuentran en la habitación. El problema pudiera ser determinar qué dijo cada persona durante la conversación. [2]

En el próximo epígrafe se describe detalladamente el problema de selección de rasgos y/o objetos.

## **1.2 Selección de rasgos y/o objetos**

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 RP están basados en la descripción de los objetos en términos de un conjunto de rasgos. [2]

En el primer grupo de problemas, lo que se enfrenta es la determinación del mejor subconjunto de rasgos para la clasificación de nuevos objetos (no clasificados). Esto conlleva la reducción del conjunto de todos los posibles rasgos (reducción de la dimensionalidad) sobre la base de las diferencias que estos rasgos presentan en cuanto a reconocer, 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. [2]

Este problema conlleva 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 la misma. [2]

De forma general el problema de selección de rasgos y/o objetos consiste en encontrar el subconjunto de rasgos que mejor describe los objetos del dominio; usualmente este subconjunto se encuentra maximizando o minimizando una función objetivo. Se reportan trabajos en la literatura para seleccionar rasgos relevantes con diferentes enfoques y técnicas, tales como: técnicas de optimización de colonias de hormigas, las de computación evolutiva, las basadas en la teoría de los conjuntos aproximados y el enfoque Lógico Combinatorio, 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. [2] [31]

Este problema se apoya en la selección del peso informacional de los rasgos y lo calcula a partir de la Teoría de Testores. [2] [31]



### 1.3 Teoría de Testores

La Teoría de Testores es una rama de la Lógica Matemática que surgió en la Unión Soviética a finales de los años sesenta. Su origen está vinculado a la utilización de métodos lógicos matemáticos para la localización de desperfectos en circuitos eléctricos que realizan funciones booleanas. [32]

Los primeros investigadores que desarrollaron estas teorías fueron, I. A. Cheguis y S. V. Yablonskii en lo concerniente a los circuitos lógicos y Y. I. Zhuravliov en la introducción de esta naciente teoría al RP. [32]. Este último aplica el concepto de testor en la solución de un problema de la Geología y presenta los primeros algoritmos para el cálculo de todos los TTs de una matriz. Es entonces que la Teoría de Testores abre toda una rama de investigaciones en la que labora un gran número de especialistas. [32]

**Definición (Zhuravliov).**- El subconjunto  $T = \{i_1, \dots, i_s\}$  de columnas de la matriz  $MA$  (y sus respectivos rasgos  $\alpha_{i_1}, \dots, \alpha_{i_s}$ ) se denomina **testor** para  $(MA_0, MA_1) = MA$ , si después de eliminar de  $MA$  todas las columnas excepto las de  $T$ , no existe fila alguna en  $MA_0$  igual a una de  $MA_1$ . [32]

Un testor  $T$  se llama irreducible (**típico**) si al eliminar cualquiera de dichas columnas deja de ser testor. [29]

#### 1.3.1 Peso informacional de los rasgos en función del concepto de Testor

El enfoque basado en la Teoría de Testores, para definir una medida de la importancia de los rasgos, se basa en la idea de que un Testor Típico (TT), según Zhuravliov, es un conjunto de columnas (rasgos) de la  $ME$  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. [2]

La idea básica consiste en lo siguiente: dada la definición de testor y TT, 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 TT (eliminando rasgos) se llega a un conjunto irreducible de rasgos, donde cada rasgo resulta imprescindible para mantener las diferencias entre las clases. Esta es la propiedad que distingue a los TT. Es natural suponer que, si un rasgo aparece en muchos conjuntos 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 los TT. [2]

Sea  $t$  el número de TTs que tiene una matriz de aprendizaje y  $t(i)$  el número de TTs en los que aparece la columna correspondiente al  $x_j$ , se dice que el peso informacional se calcula a partir de la siguiente ecuación [2]:

$$P(x_j) = \frac{t(i)}{t} \quad \forall j = 1, \dots, n, x_j \in R$$

### Ecuación 1. Peso informacional del rasgo

Haciendo uso de esta definición se han resuelto una serie de problemas de gran interés, sin embargo, la expresión anterior 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 los TT, no considera las longitudes de los testores que contienen al rasgo en cuestión, de modo que por ejemplo, si la familia de TT de un problema fuera [2]:

$$Q = \{\{x_1\}, \{x_2, x_5\}, \{x_2, x_3, x_4\}\}$$

Se tiene que:

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

Resulta natural pensar que el rasgo  $x_1$  es más relevante que  $x_3, x_4$  y  $x_5$  pues aunque sólo aparece en un TT no requiere de ningún otro rasgo para lograr diferenciar los objetos que están en clases diferentes,  $x_5$  requiere de otro rasgo y por su parte  $x_3$  y  $x_4$  requieren de otros dos; luego si se denotara por  $D(x)$  la relevancia del rasgo  $x$ , podría esperarse que:

$$D(x_1) > D(x_5) > D(x_3) = D(x_4)$$

Este análisis conduce a concluir que considerar la frecuencia de aparición de un rasgo en la familia de los TT como medida de su importancia, aunque resulta bastante natural a partir de la interpretación de un TT como un conjunto irreducible de rasgos diferenciadores, en algunos casos puede llevar a interpretaciones erróneas.

También resulta natural considerar un rasgo más importante en la medida en que es menor la longitud de los TT en que aparece.

Se asociará a cada rasgo una magnitud, que depende de estas longitudes y que se denotará por  $L(x)$  [2]:

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

### Ecuación 2. Peso del rasgo dependiendo de la longitud del Testor

Para el ejemplo anterior, se tiene que

$$L(x_1)=1, L(x_2)=0.418, L(x_3)=L(x_4)=0.333 \text{ y } L(x_5)=0.5.$$

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

Como se ha dicho,  $P(x)$  toma en cuenta sólo la frecuencia de aparición del rasgo en la familia de TT;  $L(x)$  por su parte considera únicamente la longitud de los TT en los que aparece el rasgo. [2]

Se definirá una nueva medida de la relevancia del rasgo en función de estas dos magnitudes; es decir, en general se considerará que  $D(x) = \rho(P(x), L(x))$

Considérese la siguiente expresión [2]:

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

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

Si se le concede a ambos la misma importancia se puede considerar  $\alpha = \beta = 0.5$ . Para estos valores, en el ejemplo anterior se tendrá.

$$D(x_1) = 0.666, D(x_2) = 0.542, D(x_3) = D(x_4) = 0.333 \text{ y } D(x_5) = 0.416$$

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

Existen varios algoritmos que por su estrategia de cómputo pueden clasificarse en algoritmos de escala interior y de escala exterior. Los segundos son los elegidos en el presente trabajo pues existe una investigación a la par con esta, que realiza los de escala interior.

### 1.4.1 Algoritmos de Escala Exterior

Los algoritmos de escala exterior son aquellos que realizan el cálculo de los TTs generando los elementos del conjunto potencia del conjunto de columnas de la  $MB$  en un determinado orden (usualmente utilizando un  $n$ -uplo booleano característico), de tal forma que, usando determinados criterios, el algoritmo trata de evitar el análisis de todos los subconjuntos. [29]

En la *Figura 2* se muestra un diagrama de flujo de estos algoritmos. De forma general estos algoritmos inicializan con la matriz básica, posteriormente se generan combinaciones del conjunto potencia en base dos, se comprueba si combinación generada es testor, si es testor se verifica que sea típico, si se cumple esa condición se guarda y se procede a realizar el salto TT, se verifica si se puede hacer otra posible combinación, en caso de poder se genera una nueva y en caso contrario se finaliza. Si la combinación generada del conjunto potencia no es testor, se

realiza el salto no testor y procede de la misma forma que el TT después de haber realizado el salto que se define para este.

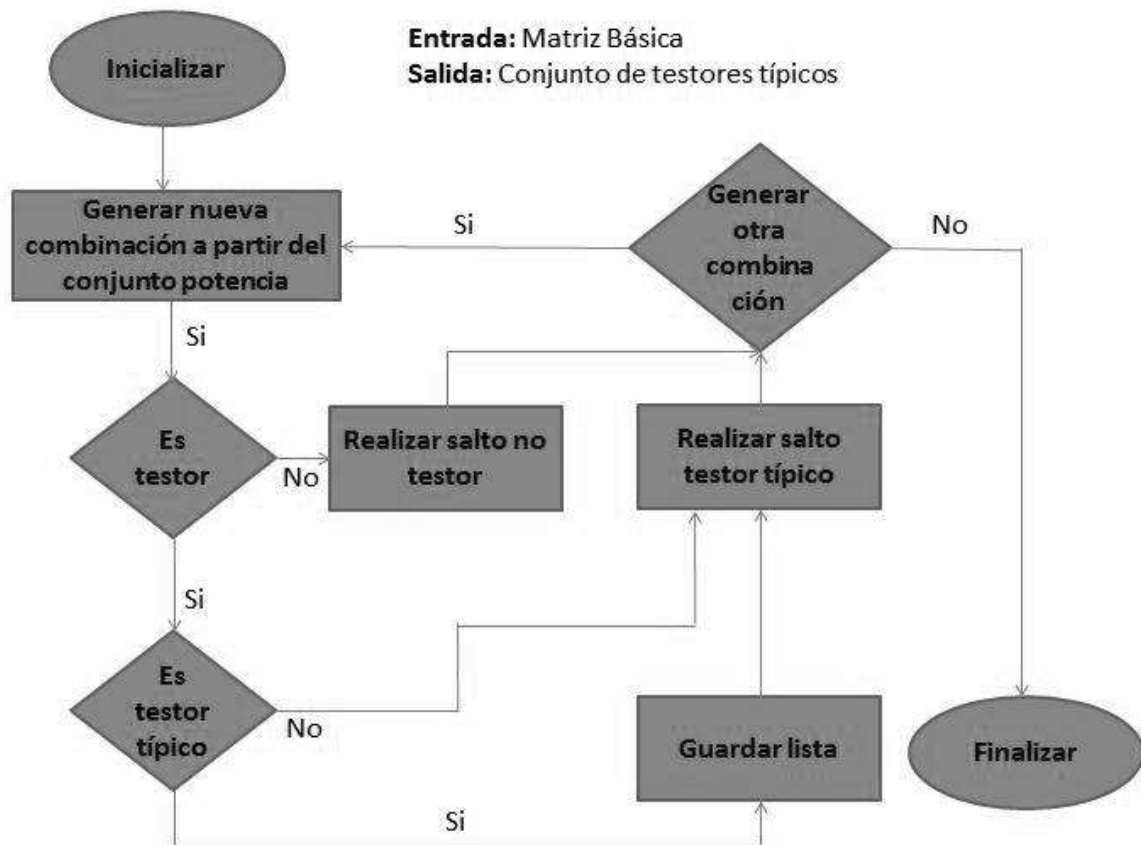


Figura 2. Diagrama de flujo algoritmos de escala exterior

### Algoritmo BT

El algoritmo BT está basado en los trabajos de especialistas soviéticos. El orden que caracteriza a este algoritmo es el generado por los números naturales de forma ascendente en su notación binaria mediante un  $n$ -uplo booleano. Este constituye un orden total sobre los elementos no nulos del conjunto potencia del conjunto de rasgos de la MA.

El algoritmo comienza entonces por el  $n$ -uplo  $(0, \dots, 0, 1)$  y procede comprobando si cada vector ( $n$ -uplo) generado es un testor o un TT. Según el resultado establecen “saltos” en el conjunto potencia. Dichos saltos se dividen en: saltos no testor y saltos testor, dependiendo si el  $n$ -uplo analizado no es testor o sí lo es. Si el  $n$ -uplo es TT se saltan todos aquellos  $n$ -uplos que son supratestores de este, y se almacena cada TT resultante en una lista (conjunto de todos los TT). Por otro lado, si el  $n$ -uplo es no testor, se observa la fila o las filas que impiden que sea testor, y se escoge aquella que tiene el 1 más a la derecha de la fila que impide que sea testor. [29] [33] [34] [35]

El algoritmo termina cuando llega al  $n$ -uplo  $(1, \dots, 1, 1)$ . Para ello se apoya en caracterizaciones de los conceptos de testor y TT y en proposiciones que definen los saltos cuando el  $n$ -uplo analizado es o no un testor. [29] [33] [34] [35]

### **Algoritmo TB**

El algoritmo TB posee iguales ventajas que el BT en cuanto al número de listas a considerar, pero resulta más ventajoso para los casos en que la longitud media de los TT es mayor que  $n/2$ , donde  $n$  es la cantidad de columnas de la matriz básica. Siendo este una mejora del algoritmo anterior. [33]

El TB procederá de manera simétrica, comenzando por la lista unitaria y generando las restantes en el orden antinatural hasta las listas nulas. [33]

### **Algoritmo BR**

El algoritmo BR evita el análisis de un mayor número de subconjuntos irrelevantes y eficientemente verifica la propiedad de testor tomando como ventaja las operaciones a nivel de bits de la computadora. El nombre del método se debe a las operaciones Binarias y a la estrategia Recursiva que éste utiliza. [24]

El método propuesto primeramente reordena las filas y columnas de  $MB$  con el objetivo de reducir el espacio de búsqueda de los TT. La fila con el menor número de 1s y el máximo número de 1s en las columnas de  $MB$  donde ésta tiene un 1 se coloca como primera fila. El reordenamiento de las columnas permite al algoritmo finalizar cuando el rasgo a ser analizado tenga un cero en la primera fila de  $MB$ , también intenta reducir la posibilidad de que los rasgos a ser analizados no sean excluyentes con una lista de rasgos y, por consiguiente, minimiza la longitud de las listas de rasgos que deben ser analizadas. [24]

La idea general del algoritmo BR es primeramente generar listas de rasgos que satisfagan la propiedad de tipicidad y luego, verificar la condición de testor. El método explora el conjunto potencia de los rasgos comenzando por el primer rasgo en la  $MB$  y genera una lista de rasgos candidata a ser TT. Una vez que ha sido generada una lista de rasgos candidata, se verifican las propiedades de testor y tipicidad. [24]

### **Algoritmo LEX**

El nombre de LEX se debe al orden lexicográfico en que son comparadas las cadenas de caracteres. La idea del algoritmo es ir construyendo listas de rasgos que posean la propiedad de tipicidad y luego, comprobar si este conjunto de rasgos constituye un testor. Cuando se obtiene un TT se producen saltos. [35]

El algoritmo empieza analizando el primer rasgo de una matriz básica  $MB$ . Un nuevo rasgo se puede incorporar a la lista si se cumple que no es excluyente con la lista, es decir, si puede

coexistir con los rasgos de la lista para formar un TT y tiene filas típicas con respecto a la lista. Posteriormente, se comprueba si el conjunto de rasgos contenidos en la lista junto con el nuevo rasgo forma un testor, verificando si no existe fila en la *MB* (tomando en cuenta únicamente esos rasgos) completa de ceros. Si se cumple que es testor, entonces estamos en presencia de un TT y se almacena. [29] [35]

Si se encontró un TT y éste contiene al último rasgo de *MB*, entonces se saltan todos sus subconjuntos consecutivos. Si no contiene al último rasgo de *MB*, para saltar todos los supraconjuntos del TT encontrado se elimina el último rasgo de la lista y se analiza si se puede incluir el próximo rasgo de *MB*. Si el rasgo analizado no se puede incorporar a la lista se prosigue el análisis con el siguiente rasgo de la *MB*. [29] [35]

Cabe indicar que previamente se realiza un ordenamiento de la *MB* que consiste en colocar como primera fila aquella que tenga la menor cantidad de unos (ordenar filas), y en esta fila encontrada, modificar las columnas de tal modo que se sitúen todos los unos a la izquierda (ordenar columnas). Si existiera más de una fila con mínima cantidad de unos se puede utilizar el concepto de Entropía Global para la selección. Así, se escoge aquella fila que tiene 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. [29] [35]

### **Algoritmo REC y CER**

El REC tiene la particularidad de que trabaja directamente con la matriz de aprendizaje, lo cual lo sitúa en desventaja con relación a los restantes algoritmos por el enorme manejo de información que tiene que realizar. [29]

El CER fue encaminado a resolver este problema y lo hizo introduciendo un nuevo orden en el conjunto potencia de los rasgos. [29]

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

Los algoritmos BT y TB, resultan ser como se observó en las descripciones anteriores los algoritmos básicos de escala exterior, a partir de los cuales se han desarrollado nuevas mejoras. Estos a su vez según la bibliografía consultada [36] [37] [38] [39] resultan ser un punto de comparación para el desarrollo de nuevos algoritmos.

En el estudio realizado en [37] sobre los distintos tiempos de ejecución de los algoritmos de escala exterior como se muestra en la tabla se pueden observar con mejores resultados al LEX y el BR.

Juegos de Datos	Clases	MB	LEX	CT-EXT	BR	NTT
Zoo(101x17)	7	14x17	0:0:00:15	0:0:0:718	0:0:00:00	34
Mushroom(8124x22)	2	30x22	0:0:00:16	0:0:0:750	0:0:00:00	292
Chess(3196x36)	2	29x36	0:2:22:16	0:8:01:67	0:0:00:12	4

Dermatology(366x34)	6	1124x34	0:25:45:7	1:43:15:6	0:0:58:22	115 556
Promoter(106x57)	2	2761x57	1:07:27:5	4:24:23:8	0:3:18:51	7 456 943
Random	- <sup>2</sup>	150x70	0:55:45:3	2:06:30:4	0:4:02:67	44 165 054
Random	- <sup>2</sup>	100x100	2:22:01:9	>20 horas	0:10:30:1	183 051 234

**Tabla 1. Tiempo del funcionamiento de los algoritmos para varias matrices básicas**

En una encuesta realizada a un grupo de personas (*ver Anexo 1*), con conocimientos en relación al RLCP, los algoritmos más conocidos son el BT, TB, LEX y BR. Se decide implementar en la presente investigación: los algoritmos BT y TB al ser punto de referencia en investigaciones de estas teorías, y los algoritmos LEX y BR por sus resultados en cuanto a tiempo de ejecución, resaltando de estos últimos su nacionalidad cubana.

### 1.5 Herramientas para la solución de problemas del Reconocimiento de Patrones

Se realizó una búsqueda de las herramientas para la solución de problemas del RP. Estas herramientas serán analizadas tomando en cuenta:

1. El trabajo simultáneo con variables cualitativas y cuantitativas. Para cada tipo 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.
2. 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.
3. 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.
4. La extensión con nuevos algoritmos para el procesamiento de la información desarrollados por los usuarios.
5. Permitir comparar los resultados sobre un mismo juego de datos al aplicar diferentes algoritmos.
6. 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.

Antes de analizar las herramientas encontradas es necesario conocer la definición de Sistema Herramienta. Un Sistema Herramienta se refiere a un complejo de programas orientados a la solución de una familia de problemas de una o más áreas específicas del conocimiento, tal que el usuario sólo requiere conocer el área específica. Si el sistema es aplicable a cualquier área del conocimiento se le denominará Universal. [40]

### **1.5.1 ALISA.**

ALISA [40] [41] fue el primer sistema construido en la URSS con el objetivo de servir de instrumento para resolver tareas planteadas en el RP. Este sistema resolvió muchos problemas de pronóstico y prospección geológica.

#### **Desventajas**

- Se concibe en base a una metodología que supone al especialista matemático como único actor posible sobre la selección del mejor algoritmo a aplicar y en la manipulación y elaboración de los datos.
- Cuenta con un conjunto de paquetes de programas y medios técnicos, que sólo para un especialista en computación es posible manipularlo con facilidad volviéndose poco accesible a todos los demás tipos de usuarios.
- A pesar de poderse 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 suponer que sea el especialista matemático el que realice la manipulación de los datos y su elaboración, se 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 [40]. Esta herramienta tiene como principal inconveniente el no estar basada en las teorías del RLCP y no permite extenderse con nuevos algoritmos.

### **1.5.2 SELECTOR.**

Fue elaborado por el Grupo de Lógica Matemática del ICIMAF de la Academia de Ciencias de Cuba. Es un sistema herramienta universal, es decir, un complejo programa orientado 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. [42].

#### **Ventajas**

- Para utilizarlo solamente se requiere tener conocimiento del área de especialidad del usuario, es decir, en Biología, Geología, Medicina, etcétera.
- Toda la fundamentación matemática del sistema está expuesta con rigor, pero sin tecnicismos, al alcance de especialistas no matemáticos. Esto permitirá hacer una real interpretación de los resultados que se alcancen, ya que se podrá saber cómo el sistema llega a los mismos.
- Permite trabajar con variables cuantitativas de cualquier tipo y con combinaciones de unas y otras.



- Pueden resolver 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.

### **Desventajas**

- No extiende nuevos algoritmos para el procesamiento de la información desarrollados por los usuarios.
- No compara los resultados sobre un mismo juego de datos al aplicar diferentes algoritmos.
- No permite el modelado de todos los tipos de variables, es decir no extiende ni agrega nuevos tipos de variables
- No permite otros criterios de comparación entre rasgos, solo los que tiene la herramienta ya implementados.

A pesar de las ventajas que nos ofrece SELECTOR, no podemos decir que es una herramienta completa, pues esta solo brinda la posibilidad de resolver problemas de selección de rasgos y objetos, por lo que no incluye los otros problemas del RLCP.

### **1.5.3 ACLAPIP**

Es un sistema de programas dirigido a la solución de problemas de clasificación, 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 variantes posibles de aplicaciones del modelo de RP basado en el peso informacional de los objetos acorde a la Teoría de Testores. [43]

ACLAPIP consta de un programa principal y 24 subrutinas. Desde el programa principal se realiza 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. [43]

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 cálculo de testores y de las diversas medidas de ellos se desprenden (como la relevancia informacional de objetos y rasgos en un problema de RP). Sin embargo, no se puede decir mucho sobre su flexibilidad pues, en términos prácticos, el usuario no podía modificar la función de semejanza considerada ni el tipo de Testor para los cuales se calcularía la relevancia de los objetos y rasgos. [1] [43]

### **Ventajas**

- Permite indicar la regla de solución que se desea utilizar en la solución del problema.

- Expresa si el problema en cuestión admite como solución la clasificación de un objeto en varias clases simultáneamente o no.

### **Desventajas**

- Ausencia de mecanismos para definir funciones de semejanzas en patrones.
- No posibilita plantear problemas bajo condiciones de comparación diferentes a las definidas por el sistema.
- No presenta mecanismos de comparación entre los resultados obtenidos por diferentes algoritmos aplicados sobre un mismo conjunto de problemas.

En la presente investigación debe tenerse en cuenta muchas de las características mejorables de la herramienta, en función de seleccionar una herramienta con más potencialidades para el RLCP.

### **1.5.4 PROGNOSIS.**

Se realizó en el año 1993 con el objetivo de ser un sistema herramienta para la resolución de problemas de RLCP con clases duras. Fue construido tratando de resolver problemas planteados por especialistas en geología, geofísica y sismología, en cuanto a crear una verdadera herramienta de trabajo y de obtención de nuevos conocimientos. Es uno de los pocos sistemas capacitados para abordar el problema de la clasificación con aprendizaje parcial de sus clases, con lo cual se coloca entre los menos que permiten toda la gama de problemas de clasificación y tareas afines. [41]

PROGNOSIS ha sido empleado en el Centro de Investigaciones Sismológicas (CENAI) del CITMA (Ministerio de Ciencia, Técnica y Medio Ambiente), en el Centro de Investigaciones para el Desarrollo del Petróleo (CIDP) entre otras instituciones. Tiene como característica que resuelve los cuatro problemas de RP [41], este sistema utiliza el algoritmo BT para calcular TTs. [34]

### **Ventajas**

- No se presupone un tipo especial de espacio de representación de 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.
- Los dominios de definición de las variables pueden ser de diversa naturaleza, cuantitativos o cualitativos; reales, booleanos, k-valentes.
- Ofrece un conjunto de funciones interconstruidas de las cuales el usuario puede seleccionar la que será usada en cada problema.

### **Desventajas**

- Carece de mecanismos para definir nuevas funciones de semejanza.

- No existe forma de que el sistema incorpore algoritmos programados por los usuarios y los pueda aplicar a conjuntos de datos previamente definidos.

PROGNOSIS ha sido aplicado con éxito en proyectos trascendentes de prospección petrolera y análisis geológico [41], pero aún dista de ser adecuado para la generalidad de los investigadores en el RPLC.

### **1.5.5 WEKA.**

Está constituido por una serie de paquetes de código abierto con diferentes técnicas de preprocesado, clasificación, agrupamiento, asociación, y visualización, así como facilidades para su aplicación y análisis de prestaciones cuando son aplicadas a los datos de entrada seleccionados. Los datos de entrada a la herramienta, sobre los que operarán las técnicas implementadas, deben estar codificados en un formato específico, denominado Attribute-Relation File Format (extensión ".arff"). [44] [45]

La herramienta permite cargar los datos en tres soportes: archivo de texto, acceso a una base de datos y acceso a través de internet sobre una dirección URL de un servidor web. En el caso de los archivos de texto podemos generarlo con cualquier editor de texto, pero al guardarlo debemos modificarle la extensión ".arff". [44] [45]

#### **Ventajas**

- Permite declarar nuevos algoritmos y trabaja con ausencia de información.
- Es una herramienta multiplataforma.

#### **Desventajas**

- No permite declarar nuevos algoritmos programados desde la perspectiva del Reconocimiento Lógico Combinatorio de Patrones.
- No cuenta con capacidad para permitir el trabajo simultáneo con variables cualitativas y cuantitativas como lo hace el RLCP.

Esta herramienta tiene como principal inconveniente el no estar basada en las teorías del RLCP por lo que no es opción para incorporar algoritmos basados en problemas de selección de rasgos y/o objetos para el cálculo de todos los TTs.

### **1.5.6 CEPAR.**

CEPAR (Entorno Cubano para el Reconocimiento Lógico Combinatorio de Patrones) v1.0.0, es un Sistema Herramienta Universal para el RLCP, tiene como objetivo fundamental apoyar en las labores cotidianas de los investigadores, docentes y estudiantes del RLCP, además proveer de una herramienta que pueda ser embebida en desarrollos propios de una investigación. [40]

Desarrollado 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. [40]

De forma nativa CEPAR maneja los tipos de rasgos:

- 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.

En la práctica esta cantidad de 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

### **Ventajas**

- Para todos los tipos de rasgos soporta la ausencia de información o valor "?".
- La herramienta permite el trabajo con variables cuantitativas y cualitativas (simultáneamente).
- Permite definir criterios de comparación y funciones de analogía entre patrones, de manera que puede extenderse con nuevos algoritmos para el procesamiento de la información desarrollados por los usuarios.
- Puede aplicar diferentes algoritmos de clasificación o de selección de rasgos sobre un mismo conjunto de datos.
- Define 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.

CEPAR, en la actualidad, no posee algoritmos que permitan trabajar con grandes volúmenes de datos, pero admite que se le puedan agregar nuevos algoritmos para tratar este tipo de problemas, permitiendo re-utilizar los algoritmos y características que actualmente posee.

### 1.5.7 Justificación de la selección

Después de describir las herramientas anteriormente planteadas se puede concluir que:

1. La presente investigación se encuentra enmarcada en las teorías y presupuestos del Reconocimiento Lógico Combinatorio de Patrones, por ello se consideran aceptadas las herramientas de este enfoque (ACLAPIP, SELECTOR, PROGNOSIS, CEPAR).
2. Las herramientas que permiten añadir nuevas funcionalidades permitirán agregar los algoritmos para el cálculo de todos los Testores Típicos de escala exterior, siendo aceptadas aquellas que además estén basadas en el RLCP (CEPAR).

Aún cuando resulta clara la selección de CEPAR como herramienta para implementar los algoritmos seleccionados con anterioridad, resulta oportuno realizar otras valoraciones:

- Las herramientas ACLAPIP, SELECTOR y PROGNOSIS poseen una antigüedad superior a los 10 años, y en este periodo de tiempo no han recibido mejoras para adaptarlas a nuevas tecnologías.
- Aunque WEKA 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 usadas

La herramienta seleccionada para la incorporación de los algoritmos, CEPAR, fue desarrollada en Java por sus facilidades como lenguaje multiplataforma, y por tal motivo es el que se adopta en este trabajo para la implementación de los mismos. De igual forma se emplea el NetBeans como Entorno de Desarrollo Integrado (IDE), al estar declarado como el IDE, en la herramienta.

#### 1.6.1 Lenguaje de programación JAVA

**Java** (*en su versión 8*) es un Lenguaje de programación orientado a objetos (POO), de alto nivel y semicompilado que funciona con una máquina virtual. Puede ser ejecutado en la mayoría de los sistemas operativos. Este lenguaje tiene un alto rendimiento ya que las optimizaciones integradas para entornos multiproceso lo hacen aún más rápido. [46]

El modelo Java para la gestión de la memoria, los procesos múltiples y la gestión de excepciones lo convierte en un lenguaje fácil de entender y eficaz para los desarrolladores nuevos. Además ofrece un entorno de aplicaciones avanzado con un alto nivel de seguridad que es idóneo para las aplicaciones de red [46]

#### 1.6.1 Entorno de Desarrollo Integrado NetBeans

**NetBeans** (*en su versión 8.0.1*) es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. 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. La plataforma ofrece servicios comunes a las aplicaciones de escritorio, permitiéndole al desarrollador enfocarse en la lógica específica de su aplicación. Entre las características de la plataforma están[47]:

- Administración de las interfaces de usuario (ej. menús y barras de herramientas).
- Administración de las configuraciones del usuario.
- Administración del almacenamiento (guardando y cargando cualquier tipo de dato).
- Administración de ventanas.

Otra característica son los paquetes que componen a NetBeans (algunos pueden ser descargados individualmente).

### **1.6.2 Herramienta CASE Visual Paradigm**

Visual Paradigm for UML (*en su versión 8.0*) que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un software. Esta herramienta permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software, además generar diversos informes a partir de la información introducida en la misma. [48]

En el presente capítulo se explica la propuesta de solución de la investigación. Esta consiste en la implementación de los algoritmos BT, TB, LEX y BR para incluirlos como extensión de la herramienta CEPAR. Se realiza una descripción de los mismos donde se muestran proposiciones y definiciones importantes, así como su pseudocódigo y posteriormente se podrá observar como quedan incluidos en dicha herramienta.

## 2.1 Descripción e Implementación

En los próximos epígrafes se describen desde el punto de vista teórico los algoritmos propuestos para la extensión de la herramienta CEPAR. Además, se describe los pasos a seguir para la implementación de los mismos.

### 2.1.1 Algoritmo BT

A continuación, se muestran un conjunto de proposiciones que se hace necesario conocer para facilitar la comprensión del algoritmo.

**Proposición 1.** La lista  $(\alpha_1, \dots, \alpha_n)$  no es una lista test cuando y sólo cuando en  $MB$  se encuentra al menos una fila  $\tilde{a} = (a_1, \dots, a_n)$  tal que  $a_j \wedge a_j = 0$  para todo  $j = 1, \dots, n$  (Aquí " $\wedge$ " representa al operador lógico conjunción).

**Proposición 2.** Sea el vector  $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n)$  una lista test típico y  $k$  el subíndice del último 1 en el vector, entonces los siguientes (después de  $\tilde{\alpha}$  en el orden natural \*)  $2^{n-k} - 1$  vectores son listas test pero no listas test típico.

**Proposición 3.** Sea el vector  $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n)$  una lista test y  $k$  el subíndice del último 1 en el vector, entonces los siguientes (después de  $\tilde{\alpha}$  en el orden natural)  $2^{n-k} - 1$  vectores son listas test pero no listas test típico.

**Proposición 4.** Sea  $\tilde{a} = (a_1, \dots, a_n)$  una fila de la matriz  $MB$  y  $k$  subíndice del último 1 en  $\tilde{a}$ . Supongamos además, que  $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n)$  no es lista test y es la primera de las listas no nulas que en el orden natural cumple la condición  $a_j \wedge a_j = 0$  para todo  $j = 1, \dots, n$ . Entonces las siguientes  $2^{n-k} - 1$  listas después de  $\tilde{\alpha}$  tampoco son listas test.

---

#### Algoritmo 1: <BT>

---

**Entrada:** Matriz básica  $MB$

**Salida:** El conjunto de todos los TTs de  $MB$

1. Se genera la primera lista de no nula de longitud  $n$  en el orden natural, esto es  $(0, \dots, 0, 1)$ .
2. Se determina en virtud de la *preposición 1*, si la lista generada es una lista de test haciendo uso de la matriz  $MB$ .
3. Si es lista test, se le aplica la *preposición 3*. Si es lista test típico se imprime la lista y se aplica la *preposición 2*. Si no es lista test, se determina la fila  $\tilde{a}$  de  $MB$ , que provoca este hecho (de no ser una única fila la causante se toma la que tenga el último 1 más a la izquierda) y se aplica la *preposición 4*.
4. Se genera la lista siguiente en virtud al *paso 3* y se regresa al *paso 2* en caso de que la lista resultante del *paso 3* no sea la unitaria, es decir  $(1, \dots, 1)$ .

### 2.1.2 Algoritmo TB

A continuación, se muestran un conjunto de proposiciones que se hace necesario conocer para facilitar la comprensión del algoritmo.

**Proposición 5.** Sea  $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n)$  una lista test típico tal que  $\alpha_{j_i} = 0, i = 1, \dots, s$ .

Entonces:

- a) cualquier lista  $\tilde{\beta} \leq \tilde{\alpha}$  no es lista test. Estas son en total  $2^{n-s} - 2$ .
- b) Las  $2^{n-j_s} - 1$  siguientes  $\tilde{\alpha}$  en el orden antinatural no son listas test.

**Proposición 6.** Sea  $\tilde{a} = (a_1, \dots, a_n)$  una fila de  $MB$ ,  $j_1, \dots, j_s$  los subíndices de todos sus unos. Supongamos que la lista  $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n)$  es la primera en el orden antinatural que cumple la condición:

- I. para todo  $j = 1, \dots, n, a_j \wedge \alpha_j = 0$ .

Entonces:

- a) todas las listas  $\tilde{\beta}$  posteriores a  $\tilde{\alpha}_0$  que cumplan:

$$\beta_j \begin{cases} = 0 & \text{si } j \neq j_1, \dots, j \neq j_s \\ \in \{0, 1\} & \text{en otro caso} \end{cases}$$

Para  $j = 1, \dots, n$ , tampoco serán listas test. En total serán  $2^{n-s} - 2$ .

- b) además la lista  $\tilde{\alpha}_0 = \bar{a} = (\bar{a}_1, \dots, \bar{a}_n)$ .
- c) Si  $j_s$  es el subíndice del último cero en  $\tilde{\alpha}_0$ , entonces las  $2^{n-j_s} - 1$  listas siguientes a  $\tilde{\alpha}_0$  en el orden antinatural tampoco serán listas test.



---

**Algoritmo 2:** <TB>

---

**Entrada:** Matriz básica  $MB$

**Salida:** El conjunto de todos los TTs de  $MB$

1. Se genera la lista unitaria de longitud  $n$ . Esto es  $(1, \dots, 1)$ .
2. Se determina en virtud de la *preposición 1* si la lista generada es una lista test, haciendo uso de la  $MB$ .
3. Si es la lista test típico, se imprime y se aplica la *preposición 5*. Si es lista test se aplica la *preposición 6*.
4. Se genera la lista siguiente a la obtenida en el *paso 3* y se regresa al *paso 2* en caso que lista resultante del *paso 3* no sea nula.

---

### 2.1.3 Algoritmo LEX

Para comprender mejor el algoritmo se hace necesario conocer algunas proposiciones y corolarios, a continuación se muestran las mismas:

**Proposición 7.** Sea  $I = [X_{j_1}, \dots, X_{j_s}]$  y un rasgo cualquiera  $X_t \notin I$ . Si existe una fila  $i$  de  $MB$  tal que cumpla:

- a)  $MB_{it} = 1$ .
- b)  $cr_i^I = 1$  siendo  $X_{j_k}$  el rasgo de  $I$  con valor unitario en la fila  $i$ ,  $1 \leq k \leq s$ .
- c)  $|F_{X_{j_k}}^I| = 1$ .

Entonces  $F_{X_{j_k}}^I = \{i\}$ , y  $X_t$  no formará parte de ningún  $TT$  junto con todos los rasgos de  $I$ .

**Proposición 8.** Sea  $I = [X_{j_1}, \dots, X_{j_s}]$  una lista no vacía de rasgos de  $MB$  y un rasgo cualquiera  $X_t \notin I$ . Si  $|F_{X_t}^I| = 0$ , entonces  $X_t$  no formará parte de ningún  $TT$  junto con todos los rasgos de  $I$ .

**Ejemplo #1:** Dada la siguiente  $MB$ :

$$\begin{array}{cccccc} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{array}$$

y la lista  $I = [X_1, X_3]$  tenemos que  $X_4$  es excluyente con dicha lista, pues cumple la *proposición 7*. Note que, en este caso, no se cumple la *proposición 8*.

**Proposición 9.** Sea  $I = [X_{j_1}, \dots, X_{j_s}]$  una lista asociada a un testor típico tal que  $X_{j_s} = X_n$ . Si existe  $X_p$  hueco de  $I$ , sea  $I' = [X_{j_1}, \dots, X_{j_k}, X_{p+1}]$ , donde  $j_1 \leq \dots \leq j_k = p - 1 < p < j_s$ . Entonces no existe ninguna lista  $\lambda$  tal que  $I \ll \lambda \ll I'$  y  $\lambda$  esté asociada a un testor típico.

**Corolario 1.** Sea  $I = [X_{j_1}, \dots, X_{j_s}]$  una lista asociada a un testor típico tal que  $X_{j_s} = X_n$ . Si no existe hueco de  $I$ , entonces no existe ninguna lista  $\lambda$  tal que  $I \ll \lambda$  y  $\lambda$  esté asociada a un testor típico.

**Corolario 2.** Sea  $I + [X]$  una lista asociada a un no testor tal que  $X = X_n$  y  $I = [X_{j_1}, \dots, X_{j_s}]$ . Si existe  $X_p$  hueco de  $I$ , sea  $I' = [X_{j_1}, \dots, X_{j_k}, X_{p+1}]$ , donde  $j_1 \leq \dots \leq j_k = p - 1 < p < j_s$ . Entonces no existe ninguna lista  $\lambda$  tal que  $I \ll \lambda \ll I'$  y  $\lambda$  esté asociada a un testor.

**Algoritmo 3:** <LEX>

**Entrada:** Matriz básica  $MB$

**Salida:** El conjunto de todos los TTs de  $MB$

1. Ordenación de  $MB$ 
  - a) Encontrar la fila con cantidad mínima de unos; de existir más de una, escoger cualquiera de ellas. Ponerla como primera fila en  $MB$ .
  - b) Ordenar las columnas poniendo indistintamente como primeras las que tengan un 1 en la primera fila.
2. Inicialización
  - a)  $I = []$ ,  $X = X_1$  ( $X$  es el primer candidato a elemento de  $I$ ).
3. Evaluación del candidato  $X$ 
  - a) Si  $I = []$  y la columna correspondiente a  $X$  tiene cero en la primera fila de  $MB$  entonces *FIN*.
  - b) Si  $X$  es excluyente con  $I$  entonces ir al paso 4; no se acepta el candidato (*proposición 1 y 2*).
  - c) Si para toda fila  $i$  de  $MB$   $cr_i^{I+X} > 0$ , entonces guardar  $I + [X]$ ; es un TT. Ir al paso 4.
  - d) Si  $X = X_n$ , entonces ir al paso 4 (*Corolario 2*).
  - e) Hacer  $I = I + [X]$ ; se acepta el candidato. Actualizar los valores de  $cr_i^I$  para todas las filas de  $MB$  y  $F_{x_t}^I$  para todo los elementos de  $X_t$  en  $I$ .
4. Selección del nuevo candidato
  - a) Si  $X \neq X_n$ , entonces sea  $j$  el índice de  $X$  en  $MB$ , hacer  $X = X_{j+1}$  e ir al paso 3.

- b) Si  $I = []$  entonces *FIN*.
- c) Si  $I = [X]$  fue un TT o  $X$  fue excluyente con  $I$ , entonces si existe el hueco  $X_p$  de  $I = [X]$ , entonces hacer  $X = X_{p+1}$  y eliminar de  $I$  todos los elementos desde  $X_p$  hasta  $X_{j_s}$  (*proposición 3*), actualizando  $cr_i^l$  para todas las filas de  $MB$  y  $F_{x_t}^l$  para cada  $X_t$  de  $I$ . Ir al paso 3. De no existir  $X_p$  entonces *FIN* (*corolario 1*).
- d) Hacer  $X = X_{j_s}$ , donde  $X_{j_s}$  es el último rasgo de  $I$  y  $I = I \setminus [X_{j_s}]$ . Actualizar  $cr_i^l$  para todas las filas de  $MB$  y  $F_{x_t}^l$  para cada  $X_t$  de  $I$ . Retornar al paso 4.

### 2.1.4 Algoritmo BR

Este algoritmo cuenta con un conjunto de proposiciones y definiciones que se hace preciso conocer para facilitar la comprensión del mismo.

**Definición 1.** Sea  $I = [X_1, \dots, X_s]$  una lista de rasgos. Se llamará *máscara de aceptación de  $I$* , denotada como  $ma_I$ , a la  $u$ -tupla binaria en la cual  $i$ -ésimo elemento es 1 si la  $i$ -ésima fila de  $MB$  tiene al menos un 1 en las columnas correspondiente a los rasgos de  $I$  y es 0, en otro caso.

**Definición 2.** Sea  $I = [X_1, \dots, X_s]$  una lista de rasgos. Se llamará *máscara de compatibilidad de  $I$* , denotada como  $mc_I$ , a la  $u$ -tupla binaria en la cual  $i$ -ésimo elemento es 1 si la  $i$ -ésima fila de  $MB$  tiene un solo 1 en las columnas correspondiente a los rasgos de  $I$  y es 0, en otro caso.

**Ejemplo #2:** Sean  $I_1 = [X_1, X_2]$ ,  $I_2 = [X_5, X_6, X_7, X_8, X_9]$  y  $I_3 = [X_1, X_2, X_8]$  listas de rasgos de una matriz básica  $MB$ . Sus máscaras de aceptación y compatibilidad correspondientes son las siguientes:

$$MB = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 & X_9 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{aligned} ma_{I_1} &= (1,1,0,0,1) \\ ma_{I_2} &= (1,1,1,1,1) \\ ma_{I_3} &= (1,1,0,1,1) \\ mc_{c_1} &= (1,1,0,0,1) \\ mc_{c_2} &= (1,1,0,0,0) \\ mc_{c_3} &= (0,1,0,1,1) \end{aligned}$$

**Proposición 10.** Una lista de rasgos  $I = [X_1, \dots, X_s]$  es un testor si y sólo si  $ma_I = (1, \dots, 1)$ .

**Definición 3.** Sea  $I = [X_1, \dots, X_s]$  una lista de rasgos y  $X \in \mathfrak{R}$ . Una fila  $p$  en  $MB$  es una *fila típica de  $X$  con respecto a  $I$*  si ésta tiene un 1 en la columna correspondiente a  $X$  y 0 en todas las columnas correspondientes a lo rasgo en  $I \setminus [X]$ .

**Proposición 11.** Sea  $I = [X_1, \dots, X_s]$  una lista de rasgos y  $X \notin I$  un rasgo de  $MB$ . La máscara de aceptación de la lista  $I + [X]$  se calcula de la siguiente forma:

$$ma_{I+[X]} = ma_I V c_x$$

**Proposición 12.** Sea  $I = [X_1, \dots, X_s]$  una lista de rasgos y  $X \notin I$  un rasgo de  $MB$ . La máscara de compatibilidad de la lista  $I + [X]$  se calcula de la siguiente forma:

$$mc_{I+[X]} = (mc_I \wedge \neg c_X) V (\neg ma_I \wedge c_X)$$

**Proposición 13.** Sea  $I = [X_1, \dots, X_s]$  una lista de rasgos y  $X \notin I$  un rasgo de  $MB$ . Si al menos una de las siguientes condiciones se cumple:

1.  $ma_{I+[X]} = ma_I$
2.  $\exists X_i \in I$  tal que  $mc_{I+[X]} \wedge c_{X_i} = (0, \dots, 0)$

Entonces,  $X$  no formará un TT con  $I$ . En este caso, se dice que  $X$  es excluyente con  $I$ .

**Ejemplo #3:**  $X_6$  es excluyente con  $I_1$  en el *Ejemplo 2*, debido a que  $X_2$  satisface  $mc_{I_1+[X_6]} \wedge c_{X_2} = (1,0,1,0,1) \wedge (0,1,0,0,0) = (0,0,0,0,0)$ . Además,  $X_8$  no es excluyente con  $I_1$ , porque  $ma_{I_1+[X_8]} \neq ma_{I_1}$  ( $I_3 = I_1 + [X_8]$ ),  $ma_{I_3} \wedge c_{X_1} = (0,0,0,0,1)$  y  $ma_{I_3} \wedge c_{X_2} = (0,1,0,0,0)$ .

**Proposición 14.** Sea  $I = [X_1, \dots, X_s]$  una lista de rasgos y  $X \notin I$  un rasgo de  $MB$ .  $I + [X]$  es un TT si y sólo si  $X$  no es excluyente con  $I$  y  $ma_{I+[X]} = (1, \dots, 1)$ .

**Definición 4.** Sean  $I = [X_1, \dots, X_s]$  una lista de rasgos,  $p$  un entero tal que  $1 \leq p \leq s + 1$  y  $X \notin I$  un rasgo de  $MB$ . Se le llamará *sustitución de  $X$  en  $I$  según  $p$* , denotado como  $sust(I, X, p)$ , a la lista  $I' = [X_1, \dots, X_{p-1}, X]$ . Si  $I = []$  entonces  $sust(I, X, p) = [X]$ .

**Definición 5.** Sean  $I = [X_{i_1}, \dots, X_{i_p}]$  y  $I' = [X_{j_1}, \dots, X_{j_q}]$  dos listas de rasgos tal que  $I \cap I' = []$ . Se le llamará *lista no excluyente de  $I$  con respecto a  $I'$* , denotada como  $noExcl(I, I')$ , a la lista compuesta por los rasgos  $X_{i_k} \in I$  tal que  $X_{i_k}$  no es excluyente con  $I'$  y  $I' + [X_{i_k}]$  no es un TT.

**Ejemplo #4:** En la  $MB$  del *Ejemplo 2*,  $noExcl(I_2, I_1) = [X_7, X_8]$ . Note que  $X_5$  y  $X_9$  no son excluyentes con  $I_1$ , pero  $[X_1, X_2, X_5]$  y  $[X_1, X_2, X_9]$  son TTs.

**Definición 6.** Sean  $I = [X_{i_1}, \dots, X_{i_p}]$  y  $I' = [X_{j_1}, \dots, X_{j_q}]$  dos listas de rasgos tal que  $I \cap I' = []$ . Se le llamará *lista típica de  $I$  con respecto a  $I'$* , denotada como  $TipL(I, I')$ , a la lista compuesta por las listas  $I' + [X_{i_k}]$  tal que  $X_{i_k} \in I$  y  $I' + [X_{i_k}]$  son TT.

**Ejemplo #5:** En la  $MB$  del *Ejemplo 2*,  $TipL(I_2, I_1) = [[X_1, X_2, X_5], [X_1, X_2, X_9]]$ .

**Definición 7.** Sean  $II = [I_1, I_2, \dots, I_t]$  una lista de listas de rasgos. Se le llamará *hueco de II*, y se denotará por  $h, 1 \leq h \leq t$ , al menor índice que cumple que  $|I_i| = |I_{i+1}| + 1, \forall i = h, h + 1, \dots, t - 1$ . Se llamará además, *eliminación del hueco de II*, y se denotará como  $ElimH(II)$ , a la lista  $[I_1, \dots, I_h]$ .

**Proposición 15.** Sea  $I$  una lista de rasgos y  $X \notin I$  un rasgo de  $MB$ . Si  $X$  es excluyente con  $I$ , entonces será también excluyente con toda la lista  $I'$ , tal que  $I \subseteq I'$  y  $X \notin I'$ .

**Algoritmo 3:** <BR>

**Entrada:** Matriz básica  $MB$

**Salida:** El conjunto de todos los TTs de  $MB$

1. Ordenando filas y columnas en  $MB$ 
  - a) Sea  $F$  el conjunto de filas que tienen el menor número de 1s.
  - b) Para cada fila  $f \in F$  obtener el número de 1s en todas las columnas de  $MB$  que contengan un 1 en  $f$ . Poner la fila con mayor número como primera fila en  $MB$ . Si existe más de una fila con el mayor número, tomar cualquiera de ellas.
  - c) Sea  $C^1(C^0)$  el conjunto de columnas con un 1 (0) en la primera fila en  $MB$ . Reordenar las columnas tal que las columnas de  $C^1$  estén a la izquierda y las de  $C^0$  estén a la derecha. Ordenar de forma descendente las columnas de  $C^1$  de acuerdo con los números de 1s. Las columnas en  $C^0$  se ordenan de la misma forma.
2. Inicialización
  - a)  $L = []$ .
  - b) Sea  $TTR$  la lista de TT,  $TTR = []$ . Note que  $TTR$  es una lista de listas.
  - c) Sea  $R$  la lista de todos los rasgos en  $MB$  y  $TL = [R]$ . Note que  $TL$  es también una lista de listas.
3. Proceso
  - a) Sea  $LR$  la última lista de rasgos en  $TL$ , esto es,  $LR = \text{último}[TL]$ .
  - b) Sea  $X$  el primer rasgo de  $LR$ .
  - c) Si  $|TL| = 1$ , entonces si la columna correspondiente a  $X(c_x)$  tiene un cero en la primera fila de  $MB$ , entonces retornar  $TTR$  y finalizar. Si no, si  $c_x = (1, \dots, 1)$ , entonces  $TTR = TTR + [[X]]$ ,  $LR = LR \setminus [X]$  e ir al paso 3b.
  - d)  $L = \text{sust}(L, X, |TL|)$

- e) Eliminar el último elemento (lista) de  $TL$ , esto es  $TL = TL \setminus [\text{último}(TL)]$ .
- f)  $LR = LR \setminus [X]$
- g)  $LP = \text{noExcl}(LR, L)$
- h)  $TTR = TTR + \text{TipL}(LR, L)$
- i) Si  $|LR| > 1$  entonces  $TL = TL + [LR]$ . Si  $|LP| > 1$  entonces  $TL = TL + [LP]$ . Si no, sea  $X_i$  el único rasgo de  $LR$ . Si  $X_i \in \text{último}(TTR)$  ó  $X_i \in LP$  entonces si existe hueco de  $TL$ , entonces  $TL = \text{ElimH}(TL)$ .
- j) Ir al paso 3.

**Nota:** Los incisos **f** y **g** pueden realizarse simultáneamente de la siguiente forma.

Para cada rasgo de  $X'$  de  $LR$ .

1. Calcular  $ma_{L+[X]}$  a partir de  $ma_L$  usando la *proposición 11*.
2. Si  $ma_{L+[X]} \neq ma_L$  (ver la condición 1 de la *proposición 13*) entonces:
  - a) Calcular  $ma_{L+[X]}$  a partir de  $mc_L$  usando la *proposición 12*.
  - b) Si  $mc_{L+[X]} \wedge c_{X_i} \neq (0, \dots, 0) \forall X_i \in L$  (ver la condición 2 de la *proposición 13*) entonces:
 

Si  $ma_{L+[X]} = (1, \dots, 1)$  entonces

Agregar  $L + [X]$  a  $TTR(L + [X])$  en un testor típico por la *proposición 14*) si no, agregar  $X'$  a  $LP$ .

---

## 2.2 Integración de los algoritmos en la herramienta

CEPAR es una herramienta implementada en Java, que presenta una arquitectura por paquetes de clases. En el paquete *cepar.core* se encuentran implementadas todas las funcionalidades que permiten el correcto modelo de un problema del RLCP, este paquete además posee todas las clases, interfaces y abstracciones necesarias para facilitar la comunicación entre los distintos paquetes relacionados a cada uno de los problemas del RP. La herramienta cuenta con un paquete destinado a la selección de rasgos y/o objetos (*cepar.featuresSelection*) en el cual se localizan las características necesarias para la correcta implementación de algoritmos para el cálculo de TTs (*ICalculateTypicalTestor* e *IOuputTypicalTestor*). La interfaz *ICalculateTypicalTestor* provee dos métodos necesarios a implementar por los algoritmos para el cálculo de todos los TTs de forma que se pueden integrar en la herramienta. El primero de estos *Inicializar* en el que recibe la *MB* y el segundo *Execute* en el cual se pasa un tipo de datos

*IOuputTypicalTestor* y se calculan todos los TTs. Esta última es la utilizada por la herramienta para recibir la información de los TTs calculados.

En la figuras 3 y 4 se puede observar el paquete *cepar.featuresSelection* con sus clases correspondientes y en la segunda como parte del aporte, se muestran los algoritmos de escala exterior propuestos incorporados a dicho paquete.



Figura 3. Paquetes de CEPAR

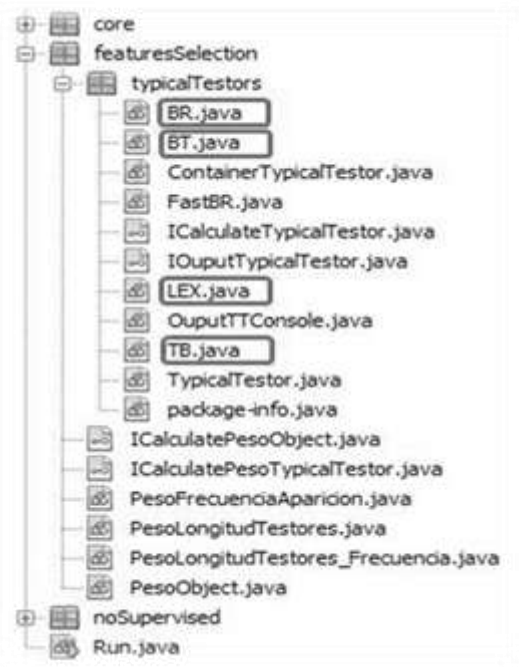


Figura 4. Paquetes de CEPAR con algoritmos incorporados

En los siguientes diagramas de clases se muestra la relación de los algoritmos implementados y de las clases incluidas en el código de CEPAR.

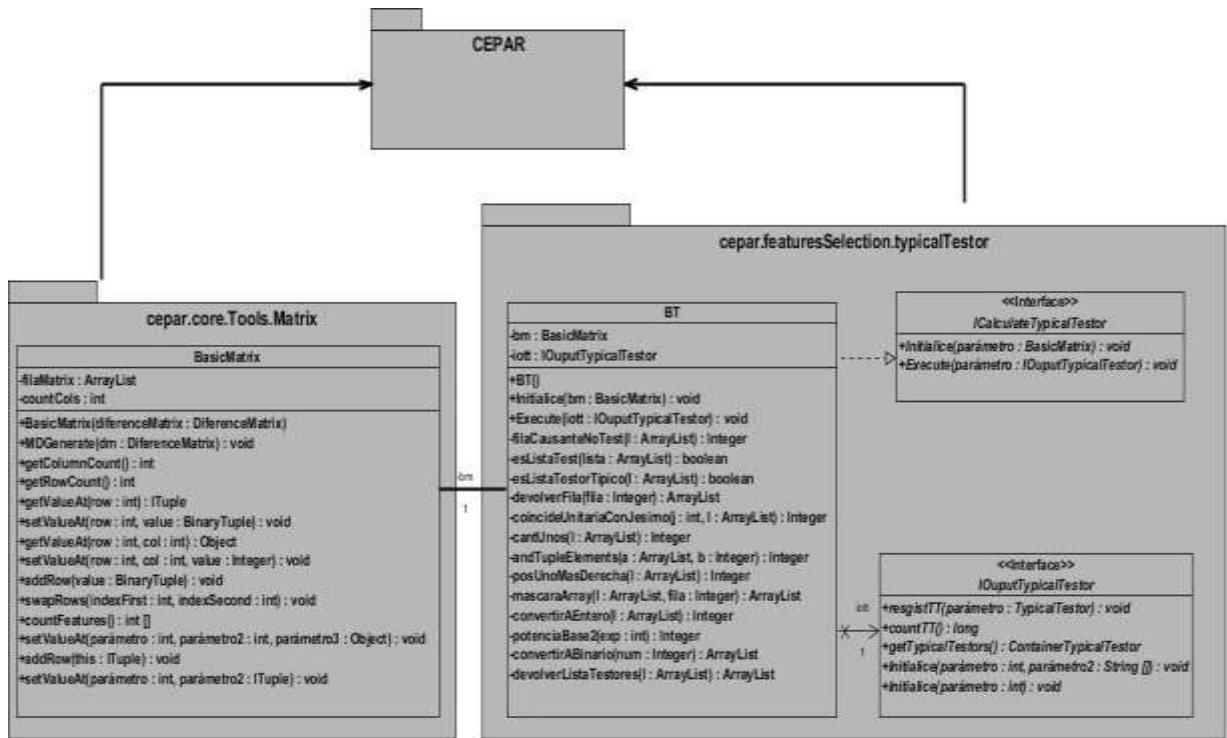


Figura 5. Diagrama de clases Algoritmo BT

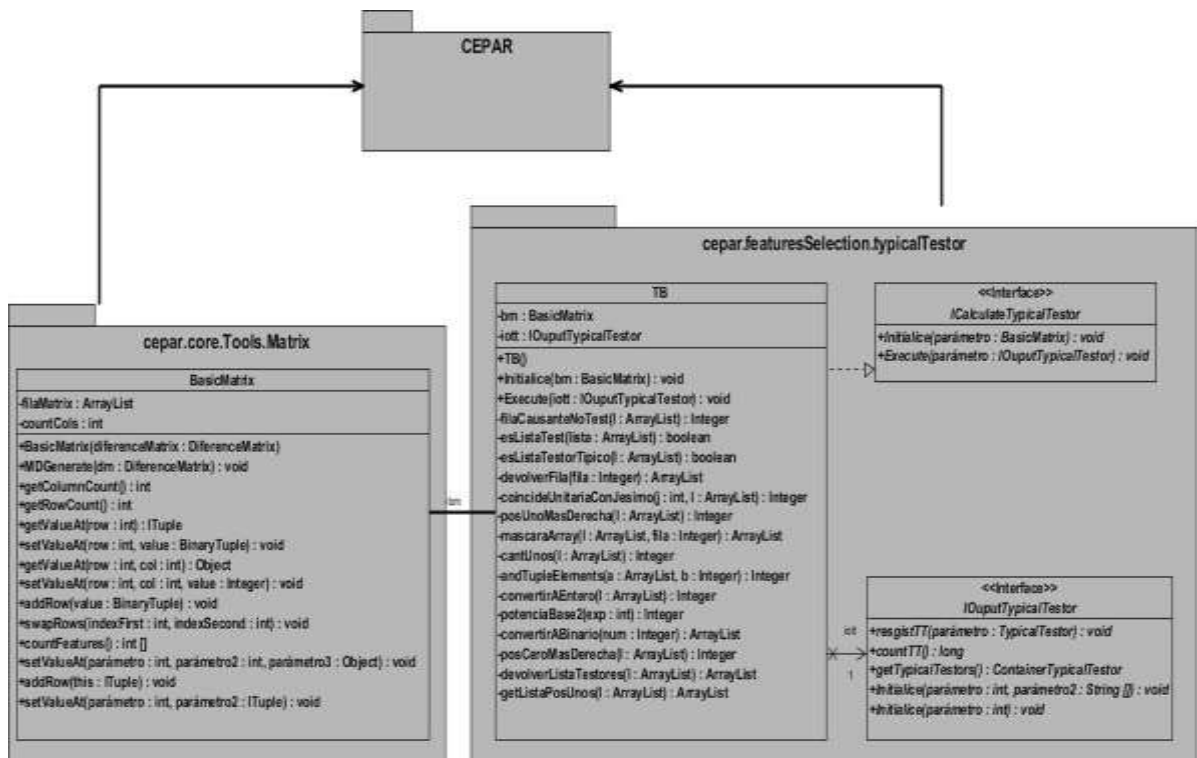


Figura 6. Diagrama de clases Algoritmo TB



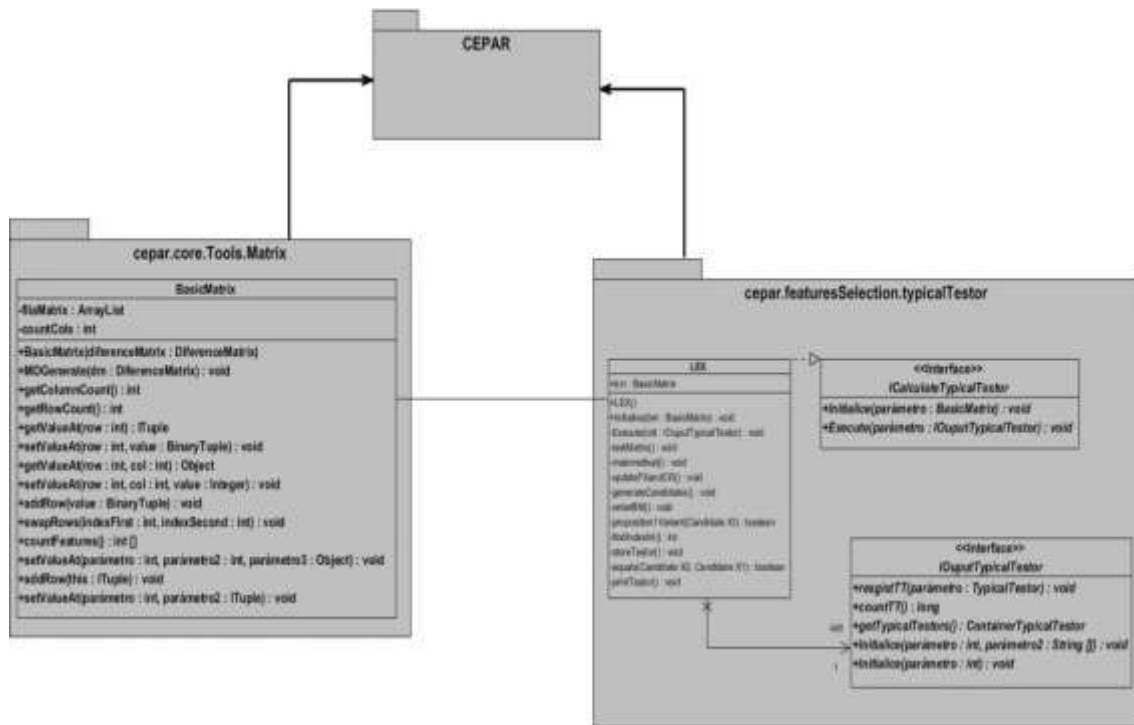


Figura 7. Diagrama de clases Algoritmo LEX

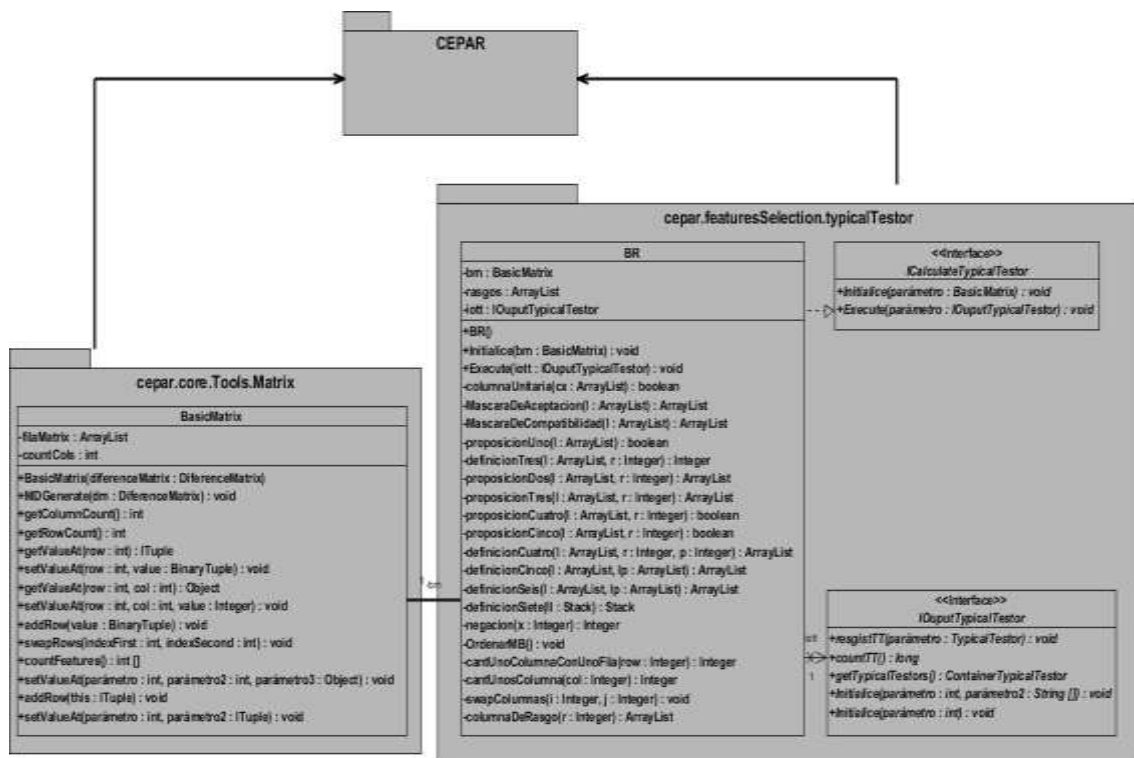


Figura 8. Diagrama de clases algoritmo BR

Al compilar nuevamente la herramienta se puede observar cómo los nuevos algoritmos implementados pueden ser elegidos desde el menú de Selección de Rasgos, Testores Típicos, Calcular Testores Típicos (Figura 9). Luego en la opción saliente ICalculateTypicalTestor se selecciona el algoritmo deseado (Figura 10) y finalmente se muestran los resultados, es decir, una

lista con todos los TTs (Figura 11), de existir en la herramienta una base de datos cargada y clasificada.

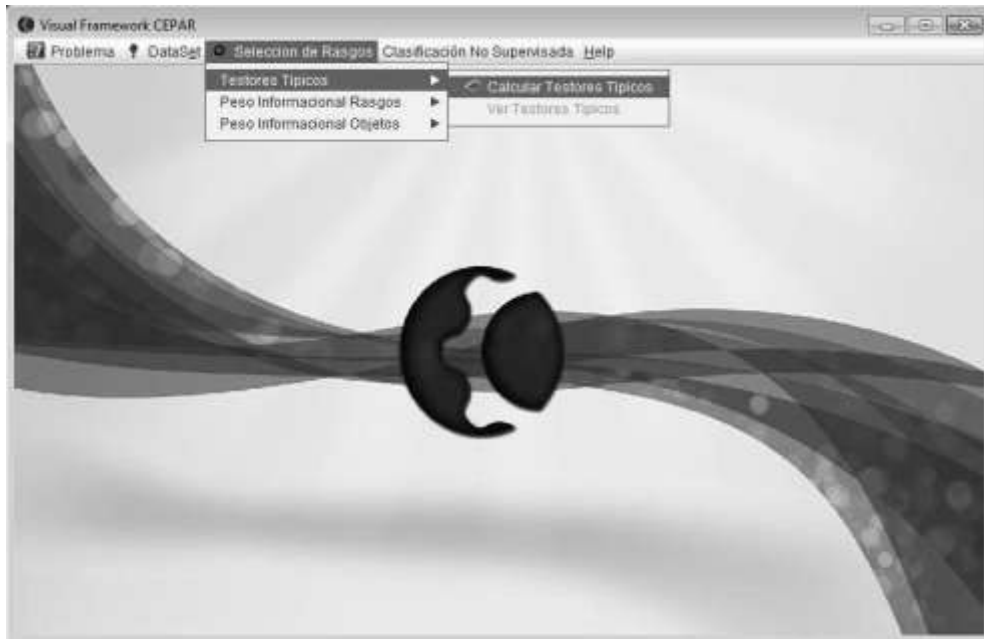


Figura 9. Selección de rasgos

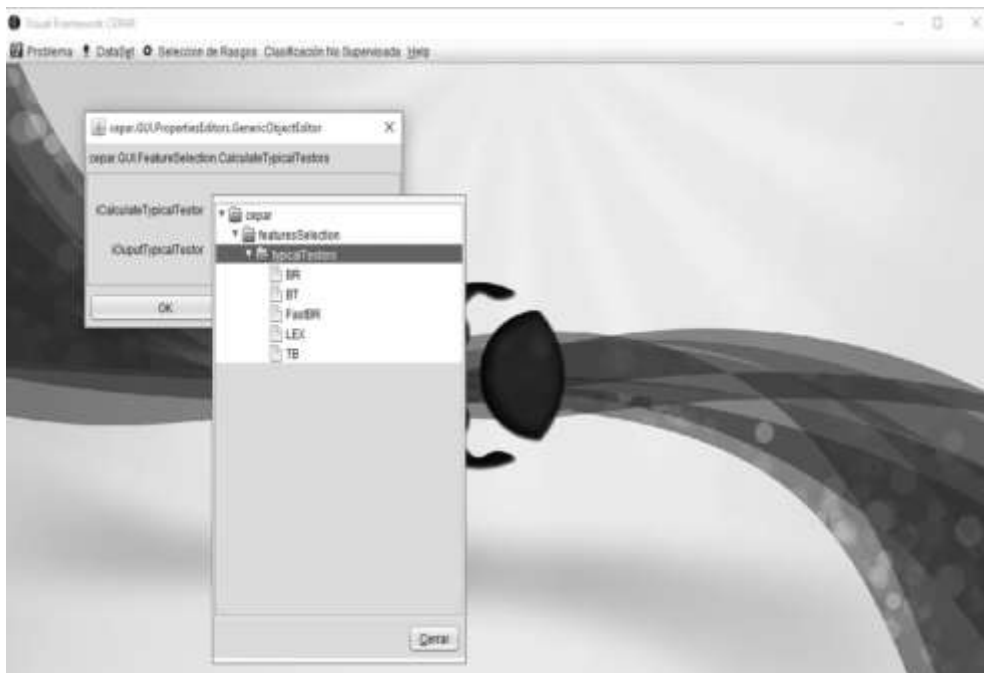
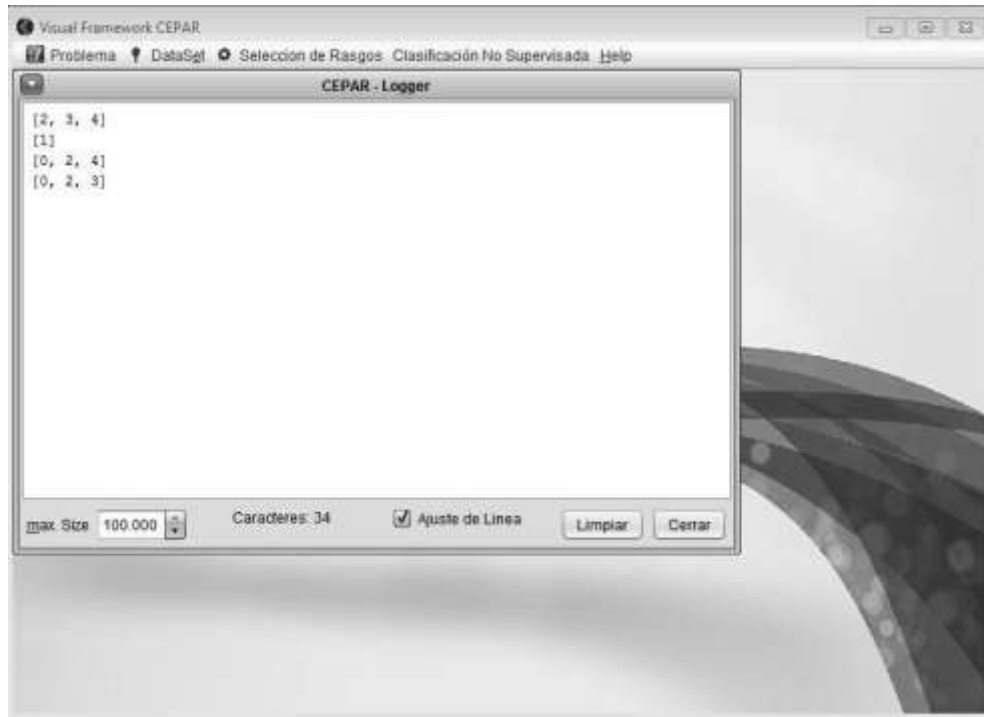


Figura 10. Selección del algoritmo a emplear



**Figura 11. Resultado del algoritmo aplicado**

Como se pudo observar en las imágenes anteriores una vez inicializada la herramienta CEPAR y cargada la base de datos con la que se realizarán los experimentos, el usuario tiene la posibilidad de seleccionar uno de los nuevos algoritmos incluidos en la herramienta.

El presente capítulo tiene como objetivo realizar las validaciones pertinentes para mostrar los resultados de la implementación de los algoritmos como extensión de la herramienta CEPAR. Para lograrlo se utilizan diferentes técnicas las cuales son: validación por expertos, pruebas de casos de estudio y validación de satisfacción a usuarios.

### 3.1 Proceso de validación de la solución

En la validación de los resultados se utilizan tres tipos de técnicas. La técnica Delphi realizada a expertos del RP para medir el nivel de concordancia, pruebas de casos de estudio para comprobar que los algoritmos implementados devuelvan todos los TTs y la técnica Iadov para obtener el grado de satisfacción de los usuarios.

A continuación se muestra un diagrama con los procesos de validación que se realizaron en la presente investigación:

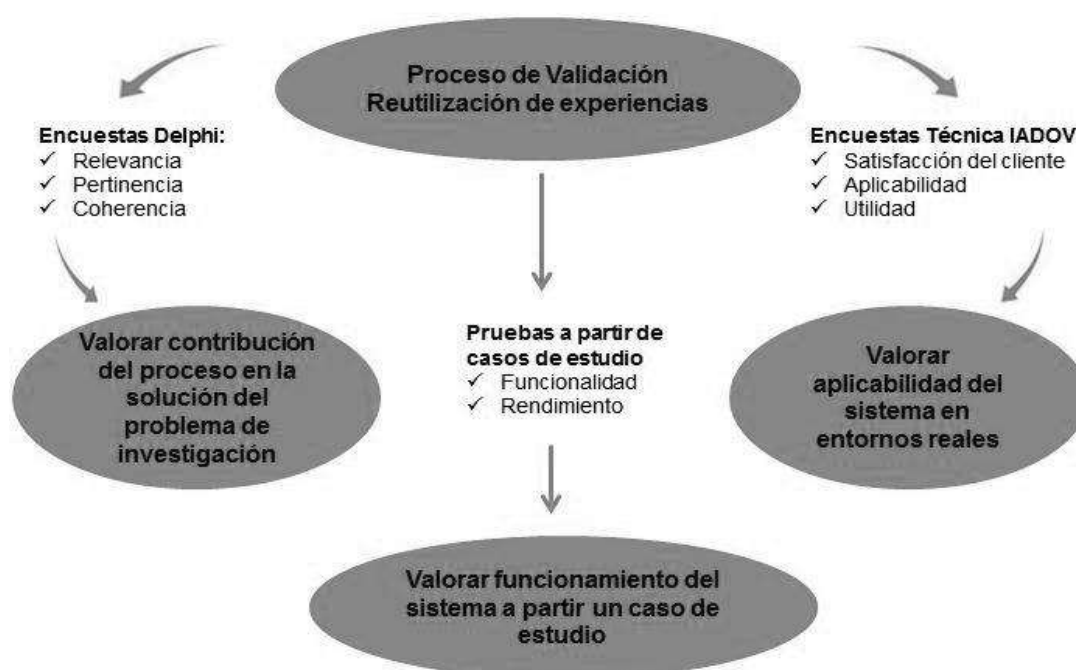


Figura 12. Procesos de validación

### 3.2 Validación por expertos

La técnica Delphi [49], es 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.

Las características que definen y configuran al método Delphi como una técnica grupal relevante, son las siguientes:

**Iteratividad:** En más de una ocasión los participantes en el proceso tienen que emitir su opinión, a través de rondas sucesivas, por lo que el proceso finaliza en el momento 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:** Permite eliminar el efecto negativo que ejercen algunos participantes dominantes. Por ello, ningún miembro del grupo a entrevistar debe conocer las respuestas del resto, lo cual puede implicar, que los participantes no conozcan quienes son los otros componentes del grupo. Para lograr el anonimato 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 en cada ronda o previamente al inicio de la siguiente. Además, 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.

Landeta [50] plantea que 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”.

Según los estudios de Landeta [50] respecto al número de expertos a participar, señalan que la cantidad de expertos debe ser mayor o igual a 5 y no mayor que 30.

Esta técnica ha sido utilizada en la presente investigación con el objetivo de validar el problema planteado y de igual forma para comprobar la pertinencia de la propuesta de solución.

### **3.2.1 Proceso de selección de los expertos**

Para la selección del grupo de expertos, primeramente se tuvo en cuenta el conocimiento que tienen en el área donde se inserta la investigación sobre la base de su currículum personal. De esta

manera se identificaron 15 posibles candidatos de los cuales se descartaron 7 por falta de disposición a participar. Quedando 8 del total inicial, después, se realizó una valoración sobre el nivel de experiencia que tienen, evaluando de esta forma los niveles de conocimientos que poseen sobre la materia.

Para determinar el nivel de experiencia de los expertos se realizan dos preguntas, una primera pregunta de autoevaluación de los niveles de información y argumentación que tienen sobre el tema en cuestión. En esta pregunta se les pide 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										

**Tabla 2. Tabla para el procesamiento del grado de 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:

$$K_c = n(0,1) = \frac{n}{10}$$

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 agrupamiento o fundamentación		Alto	Medio	Bajo
1	Su 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. Procesamiento de los aspectos que pueden influir en el 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 agrupamiento o fundamentación		Alto	Medio	Bajo
1	Su experiencia teórica	0.30	0.20	0.10
2	Su experiencia práctica	0.50	0.40	0.20
3	Bibliografía nacional consultada	0.05	0.40	0.20
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 intuición	0.05	0.05	0.05

**Tabla 4. Tabla 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:

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

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 calcular 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:

$$K = 0.5(K_c + K_a)$$

Donde:

$K$ : Coeficiente de Competencia

$K_c$ : Coeficiente de Conocimiento

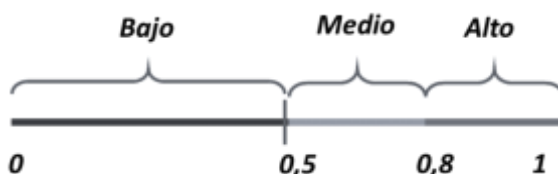
$K_a$ : Coeficiente de Agrupamiento

Los resultados se analizan de la manera siguiente 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 13. Coeficiente de Competencia**

Después de obtener los resultados (ver Tabla 5), se determinó la selección de aquellos que tuviesen resultados satisfactorios de desempeño en los valores del coeficiente de competencia (alto) para estudiar, analizar, 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 los expertos**

N°	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	Máster	Profesor Asistente	8
3	Si	Máster	Profesor Asistente	13
6	Si	Ingeniero	Profesor Asistente	9
7	Si	Doctor	Profesor Titular	30
8	Si	Ingeniero	-	3

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



### 3.2.2 Elaboración del cuestionario

Una vez seleccionados los expertos, se prosigue con la elaboración de la encuesta de validación. La misma está integrada por 5 preguntas en las cuales se recoge toda la valoración respecto al problema planteado y a la propuesta de solución, resaltando lo positivo, negativo e interesante con el objetivo de poder lograr una propuesta lo más completa posible. El cuestionario fue creado de forma tal que las respuestas se categorizaran en Muy Adecuado (A1); Bastante Adecuado (A2); Adecuado (A3); Poco Adecuado (A4) y No adecuado (A5). Todos son equivalentes a un peso numérico de 5, 4, 3, 2, 1 respectivamente. Consultar (Anexo 2), para conocer en detalle el cuestionario de validación.

### 3.2.3 Análisis y procesamiento de los resultados

El establecimiento de la concordancia de los expertos mediante el *coeficiente de Kendall*, posibilita definir el nivel de concordancia entre los expertos. Por lo que un perfecto acuerdo entre ellos dará mayor validez a la propuesta.

Se muestran los resultados brindados por los expertos, reflejando la puntuación de las respuestas en base a 5 puntos (ver Tabla 7).

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

**Tabla 7. Resultado de las encuestas a expertos**

Una vez establecida la puntuación de las respuestas dadas por los expertos encuestados, se hace uso de la fórmula para el cálculo del Coeficiente de Kendall ( $W$ ), la cual plantea que:

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

#### **Ecuación 3. Fórmula del Coeficiente de Kendall**

- $K$  es el número de expertos que intervienen en el proceso de validación,  $k = 5$ .
- $N$  cantidad de aspectos a validar,  $N = 5$ .
- $S$  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,  
 $R_j = 113$ .
- $\bar{R}_j$  es la media de los rasgos y se determina mediante la fórmula:

$$\bar{R}_j = \frac{\sum_{j-i}^n R_j}{N} ; \bar{R}_j = \frac{113}{5} = 22.6$$

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

$$\begin{aligned} S &= (20 - 22.6)^2 + (24 - 22.6)^2 + (24 - 22.6)^2 + (21 - 22.6)^2 + (24 - 22.6)^2 \\ &= 6.76 + 1.96 + 1.96 + 2.56 + 1.96 \\ &= 15.2 \end{aligned}$$

Después de obtener los valores, se puede calcular el Coeficiente de Kendall (W) a través de la fórmula:

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

$$W = \frac{12 * 15.2}{5^2(5^3 + 5)} = \frac{182.4}{25 * 120} = \frac{182.4}{3000} = 0.060$$

El resultado obtenido **0.060** expresa el grado de acuerdo entre los 5 expertos al dar un orden evaluativo a las preguntas sometidas a valorar.

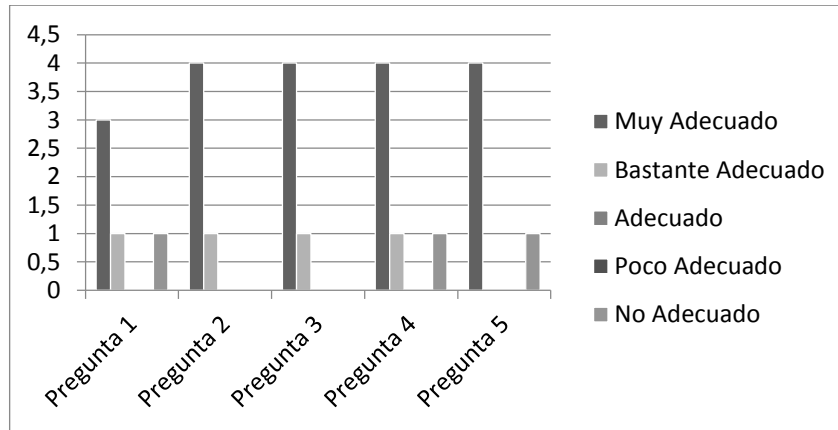
Luego se procede con el cálculo del **Chi-Cuadrado** con el objetivo de poder ver si existe concordancia entre los expertos encuestados:

$$X^2 = K(N - 1)W ; X^2 = 5 * (5 - 1) * 0.060 = 1.2$$

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 probabilidad de error de 0,01.

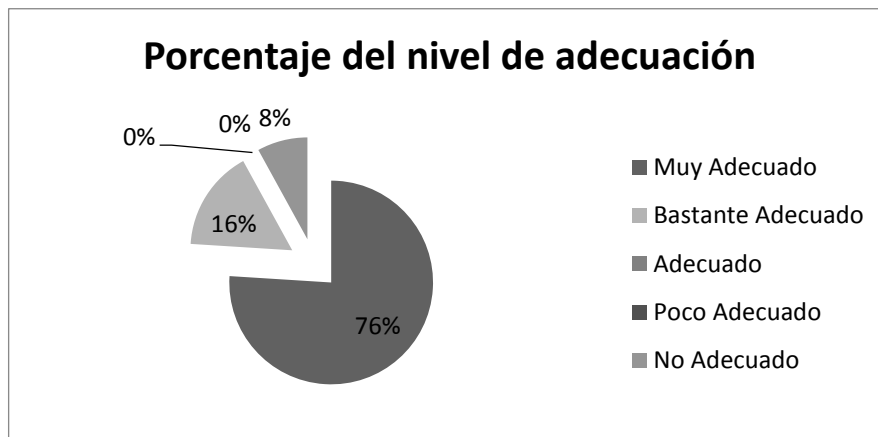
Si  $x_{real}^2 < x_{(\alpha, N-1)}^2$ , entonces hay concordancia entre los expertos.  $x_{real}^2 = 1.2$  es menor que  $x_{(0.01; 4)}^2 = 13.27$ , por tanto, existe concordancia entre los expertos, lo que significa que están de acuerdo con la incorporación de los algoritmo como extensión de la herramienta seleccionada.

A continuación se muestra una gráfica que refleja el resultado de la superioridad de la opción Muy Adecuado sobre las demás, donde se demuestra además la validez de la propuesta presentada.



**Figura 14. Gráfica de adecuación de preguntas**

Una vez concluido el proceso de validación y registrado cada uno de los datos se puede llegar a la conclusión de que el 76% de las respuestas fueron categorizadas de Muy adecuada, el 16% de Bastante adecuada y el 8% de Adecuada, 0% respuestas detectadas de Poco adecuada y 0% No adecuada. En la siguiente gráfica se muestra la tabulación de los porcentajes.



**Figura 15. Porcentaje del nivel de adecuación**

### 3.3 Resultado experimental a partir del desarrollo de casos de estudio

El objetivo de este epígrafe es comprobar que los algoritmos desarrollados funcionen correctamente, y que la incorporación de mismos para el cálculo de todos los TTs en la herramienta CEPAR facilita la selección del mejor algoritmo para la resolución de un problema dado.

Todas las pruebas se llevan a cabo en un procesador Intel Pentium Dual Core a 2,4 GHz, con 4 GB de RAM para garantizar una comparación equitativa. En esta comparación se utilizan cuatro

conjuntos de datos obtenidos a partir de la UCI Machine Learning Repository<sup>1</sup>. Para cada uno, se generaron las matrices básicas relativas al resultado de procesarlas tomando como:

- 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):

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

- Umbral de semejanza: Media de las máximas semejanzas:

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

- Criterio de agrupamiento  $\beta_0$  – *Compacto*.

En la herramienta CEPAR, se encuentran implementadas todas las configuraciones antes expuestas para la generación de la matriz básica que sirve de entrada para el cálculo de todos los testores típicos, de igual forma la herramienta también cuenta con la implementación del algoritmo Fast-BR, que sirve para comprobar que los TTs generados son los esperados.

Juegos de Datos	Clases	MB	BT	TB	LEX	BR	NTT
Zoo(101x17)	24	17x17	609 ms	567 ms	203 ms	62 ms	13
Mushroom(8024x22)	2438	13x22	5750 ms	5289 ms	63 ms	47 ms	4
Flags(194x30)	74	1396x30	+2hrs	+1hr 30 min	434656 ms	6516 ms	101 207
Dermatology(366x34)	192	2827x34	-----	-----	+1hr	640897 ms	67 578 883

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

Los resultados de las distintas ejecuciones al aplicar los algoritmos a los juegos de datos, permiten llegar a la conclusión que algoritmo BR es el que menos tiempo de ejecución demuestra.

### 3.4 Validación de satisfacción de potenciales usuarios

Con la finalidad de conocer el grado de satisfacción de potenciales usuarios respecto a la propuesta, se aplicó la técnica de ladov. Esta técnica se basa en la aplicación de un cuestionario (ver Anexo 2) en el que se insertan tres preguntas cerradas y dos abiertas, además de preguntas secundarias. El cuadro lógico de ladov (ver

Tabla 9) establece la relación entre las tres preguntas cerradas que están dirigidas a conocer la satisfacción respecto al objetivo que motiva la encuesta. [51]

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

La técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen tres cuestionamientos cerrados y el análisis posterior de dos cuestiones abiertas, que se intercalan dentro del cuestionario y cuya relación el encuestado desconoce. La relación entre las preguntas cerradas se establece a través del denominado "Cuadro Lógico de ladov". [52]

		<b>P 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?</b>								
		No			No sé			Si		
<b>P 9. ¿Satisface sus necesidades como (estudiante   investigador   docente) una herramienta con varios algoritmos para el cálculo de todos los Testores Típicos?</b>		<b>P 11. ¿Si usted necesitara desarrollar una aplicación en la que se lleve a cabo el cálculo de todos los Testores Típicos usaría esta propuesta?</b>								
		Si	No sé	No	Si	No sé	No	Si	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 de ladov modificado por los autores**

El número resultante de la interrelación de las tres preguntas indica la posición de cada cual en la escala de satisfacción, para estas actividades responde a la siguiente estructura:

<b>Escala de satisfacción</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 tabla anterior es necesaria para calcular el Índice de Satisfacción Grupal (ISG), el cual se calcula a través de la siguiente fórmula:

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

#### Ecuación 4. Índice de Satisfacción Grupal

En esta fórmula las variables A, B, C, D, E, representan el número de sujetos con índices individual 1; 2; 3 ó 6; 5; 4 en cada posición de la escala y N representa el número total de sujetos del grupo. Se denota por A para los de la posición 1, por B para los de la 2, D para la 4, E para la 5 y por C para el número de encuestados en las escalas 3 y 6.

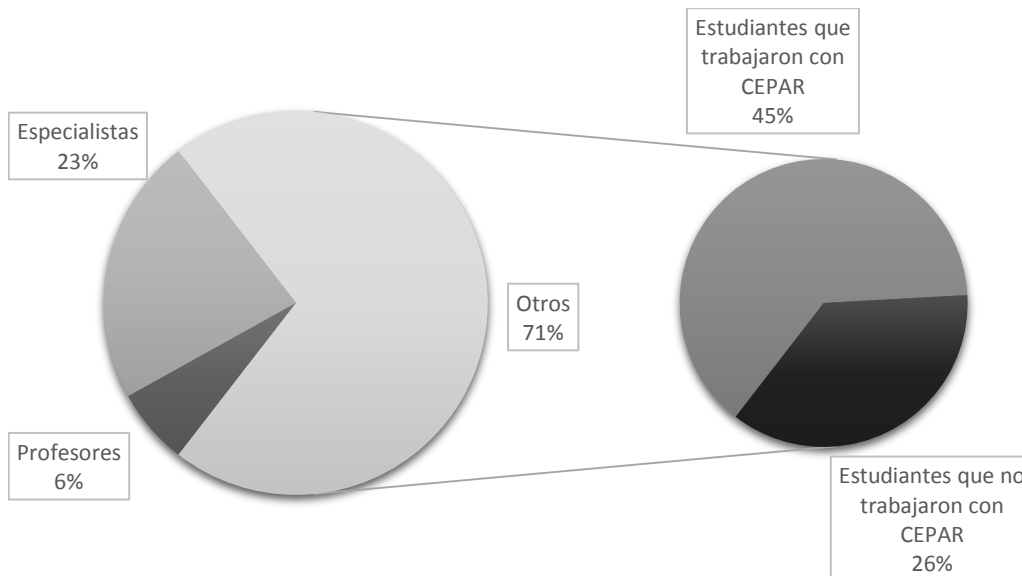
El ISG arriba valores entre +1 y -1. Los valores que se encuentran comprendidos entre -1 y -0.5 indican insatisfacción; los comprendidos entre -0.49 y +0.49 evidencian contradicciones y los que caen entre +0.5 y +1 indican que existe satisfacción:



Tabla 11. Intervalos de satisfacción

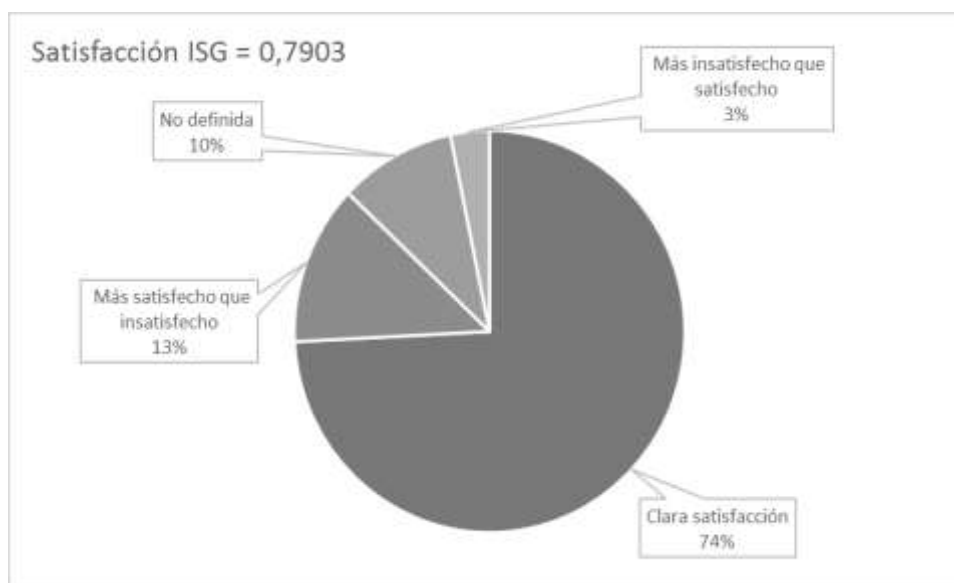
#### 3.4.1 Análisis de los resultados

La encuesta (*ver Anexo 3*) fue aplicada a 35 personas, de ellas 4 no se tuvieron en cuenta porque no poseen conocimientos sobre el Reconocimiento Lógico Combinatorio de Patrones, quedando una muestra de 31 personas en total. De las cuales nueve eran especialistas o profesores, y el resto estudiantes de la Facultad 2 de la Universidad de las Ciencias Informáticas que cursaron (o cursan) la asignatura optativa Reconocimiento de Patrones. Es necesario señalar que los estudiantes matriculados en la asignatura, en el segundo semestre del curso 2015-2016 se apoyaron en la herramienta CEPAR para desarrollar su proyecto final.



**Tabla 12. Características de los encuestados de ladov**

El ISG arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; 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 obtenido fue de 0.7903 lo que indica satisfacción con respecto a la propuesta. El resultado del nivel de satisfacción se muestra en la siguiente figura.



**Figura. 1 Resultados de la aplicación de la técnica de ladov y valor del índice de satisfacción grupal (ISG)**

### 3.5 Resultados de las pruebas y validaciones

Las pruebas y validaciones realizadas a la herramienta con los algoritmos incorporados arrojaron resultados satisfactorios; muestra de ello, la validación de la propuesta a través del método Delphi donde se comprobó la concordancia de los expertos al considerar necesario una plataforma, para

la incorporación de varios algoritmos para el cálculo de todos los TTs. Además, se pruebas con casos de estudio a los algoritmos implementados demostrándose que los resultados de la propuesta de solución son los esperados y se calculó el Índice de Satisfacción Grupal por parte de los posibles usuarios y se obtuvo como resultados 0.7903 lo cual demuestra según el intervalo de satisfacción que los usuarios están satisfechos con la propuesta de solución.



---

## Conclusiones

---

Se describe el resultado de una investigación que comenzó con el objetivo de obtener una herramienta que facilitará el trabajo de selección y comparación de algoritmos para el cálculo de todos los testores típicos. Con dicho propósito se abordaron temas relacionados con el RLCP, la Teoría de Testores, los algoritmos de escala exterior para el cálculo de todos los Testores Típicos y los Sistemas Herramientas existentes para el RLCP. Además, se concluye que:

- Debido a que los algoritmos para el cálculo de todos los TTs se encuentran dispersos la comparación de los mismos resulta compleja, haciendo el trabajo del investigador engorroso; con el uso de la aplicación se le facilita al investigador realizar comparaciones entre los mismos, apoyando así la toma de decisiones.
- Se realizó una caracterización de los algoritmos de escala exterior para el cálculo de todos los TTs, de esta manera se logró la selección de los más destacados en las investigaciones prácticas.
- Al realizar una comparación entre los sistemas herramientas para la solución de problemas del RLCP, CEPAR posee características que permiten su extensión con nuevos algoritmos atendiendo a los criterios de selección empleados.
- Se implementaron los algoritmos de escala exterior para el cálculo de todos los TTs para problemas de selección de rasgos y/o objetos, en la herramienta CEPAR obteniéndose los resultados esperados después de evaluarse los casos de estudio.
- Se validó la propuesta de solución mediante el método Delphi de consulta a expertos demostrando la concordancia entre los mismos y se realizó la técnica IADOV obteniendo como resultado la satisfacción de los usuarios con el desarrollo de la propuesta.

---

## Recomendaciones

---

Se recomienda la incorporación de otros algoritmos de escala exterior para el cálculo de todos los TTs en la herramienta CEPAR, resaltando entre ellos los que utilicen técnicas de programación paralela.

Ampliar la gama de algoritmos en la herramienta incluyendo aquellos que tienen en cuenta que las clases de los objetos pueden ser difusas.

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.

Implementar algoritmos conceptuales que utilizan los TTs, según el RLCP, en la herramienta incrementando el alcance de la actual propuesta.

---

## Referencias

---

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*. Instituto Politécnico Nacional, 2009.
2. Ruíz-Shulcloper, J. *Reconocimiento Lógico Combinatorio De Patrones: Teoría Y Aplicaciones*. 2009.
3. Ruíz-Shulcloper, J., Carrasco-Ochoa, J.A. y Martínez-Trinidad, J.F. Reconocimiento de patrones: conceptos y metodología. En: *Serie Azul*, CENATAV. 2013, ISSN 1098-6596. DOI 10.1017/CBO9781107415324.004.
4. Jain, A.K., Duin, R.P.W. y Mao, J. Statistical Pattern Recognition: A Review. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 2000, Vol. 22, núm. 1, p. 4-37. ISSN 01628828. DOI 10.1109/34.824819.
5. Fukunaga, K. *Introduction to Statistical Pattern Recognition*. 2nd. 1990. ISBN 0-12-269851-7.
6. Martínez-Sánchez, N. y Reyes-González, Y. *Algoritmos Conceptuales: Una perspectiva para el modelado del estudiante en los Sistemas Tutoriales Inteligentes*. 2014.
7. Espinosa-Alba, I. y Gutiérrez-Rosa, L.T. *La minería de datos como soporte a la toma de decisiones estratégicas de las organizaciones*. 2010.
8. Quintos-Vázquez, L.E. *Reconocimiento de Patrones, el enfoque Lógico Combinatorio*. Instituto Politécnico Nacional Escuela Superior de Física y Matemáticas, 2009.
9. María-Seijas, L. *Reconocimiento de patrones utilizando técnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos*. 2011.
10. Guzmán-Arenas, A. y Martínez-Trinidad, J.F. The logical combinatorial approach to pattern recognition, an overview through selected works. En: *Pattern Recognition*. 2001, Vol. 34, p. 741-751.
11. Lazo-Cortés, M., Ruíz-Shulcloper, J. y Alba-Cabrera, E. An overview of the evolution of the concept of testor. En: *Pattern Recognition*. 2001, Vol. 34, núm. 4, p. 753-762. ISSN 00313203. DOI 10.1016/S0031-3203(00)00028-5.
12. Ruiz-Shulcloper, J. y Fuentes-Rodríguez, J. Modelo cibernético para el análisis de la delincuencia juvenil. En: *Ciencias matemáticas*. 1981, Vol. 2, núm. 1, p. 141-153. ISSN 0256-5374.
13. Ortiz-Posadas, M.R. y Vega-Alvarado, L. *Análisis de una muestra de pacientes con labio-*

*paladar hendido usando un algoritmo de tipicidad y contraste*. 2001.

14. Ortíz-Posadas, M.R. et al. *El concepto de analogía en medicina. Una función de semejanza para pacientes con fisuras de paladar*.

15. Ortíz-Posadas, M.R., Maya-Behar, J. y Lazo-Cortés, M. Evaluación de la cirugía de labio y paladar hendido con el enfoque lógico combinatorio de la teoría de reconocimiento de patrones. En: *Cuaderno de Ingeniería Biomédica*. 1998, Vol. 14, p. 7-22.

16. Tobergte, D.R. y Curtis, S. *Aplicación de técnicas de reconocimiento de patrones con enfoque lógico combinatorio en la clasificación de un complejo de géneros de hongos patógenos en la caña de azúcar*. ISBN 9788578110796. DOI 10.1017/CBO9781107415324.004.

17. Gómez-Herrera, J.E. et al. Pronóstico gasopetrolífero en la asociación ofiolítica cubana aplicando modelación matemática. En: *Geofísica Internacional*. 1994, Vol. 33, núm. 3, p. 447-467.

18. Torres-Soto, M.D. et al. *Testores en la determinación de factores de riesgo:TRALI*. 2013.

19. Pico-Peña, R., Álvarez-Gómez, L. y Cotilla-Rodríguez, M. *Zonación sísmica de la isla de Cuba mediante algoritmos de clasificación lógico combinatorio*.

20. Caballero-Mota, Y. et al. Un nuevo algoritmo de selección de rasgos basado en la teoría de los conjuntos aproximados. En: *Revista Facultad de Ingeniería Unoversidad de Antioquia Colombia*. 2007, núm. 41, p. 132-144.

21. Rojas-Delgado, J. Selección de rasgos en muestras citológicas usando información heurística. En: *Revista Cubana de Ciencias Infórmaticas*. 2016, Vol. 10, núm. 2, p. 73-88.

22. Castañeda-Cao, O. y García-Lorenzo, M.M. Análisis y propuesta de selección de rasgos para el Reconocimiento de Expresiones Faciales. En: *Revista Cubana de Ciencias Infórmaticas*. 2015, Vol. 9, núm. 1, p. 45-62.

23. Ruíz-Shulcloper, J. Acerca del surgimiento del Reconocimiento de Patrones en Cuba. En: *Revista Cubana de Ciencias Infórmaticas*. 2013, Vol. 7, núm. 2. ISSN 2227-1899.

24. Pons-Porrata, A. y Lias-Rodríguez, A. *Un nuevo algoritmo de escala exterior oara el cálculo de los testores típicos*. 2009.

25. Ortíz-Posadas, M.R., Martínez-Trinidad, J.F. y Ruíz-Shulcloper, J. A new approach to differential diagnosis of diseases. En: *Bio-Medical Computing*. 1996, Vol. 40, p. 179-185. ISSN 00207101. DOI 10.1016/0020-7101(95)01143-9.

26. Pons-Porrata, A., Ruiz-Shulcloper, J. y Berlanga-Llavori, R. *A Method for the Automatic Summarization of Topic-Based Clusters of Documents*. 2003. 2003.

27. Coello-González, S. y Hernández-Leónn, R.A. *El Paradigma Cuantitativo de la Investigación*

*científica*. 2002. ISBN 9591603436.

28. Carrasco-Ochoa, J.A. y Martínez-Trinidad, J.F. Reconocimiento de patrones. En: *Komputer Sapiens*. 2011, Vol. II.

29. Pons-Porrata, A. y Santiesteban-Alganza, Y. LEX : Un nuevo algoritmo para el cálculo de los testores típicos. En: *Revista Ciencias Matemáticas*. 2003, Vol. 21, núm. 1.

30. Guerra-Gandón, A., Vega-Pons, S. y Ruiz-Shulcloper, J. Algoritmos de agrupamiento conceptuales: un estado del arte. En: *Serie Azul , CENATAV*. 2012,

31. Reyes-González, Y. *Modelo para la adaptación de las soluciones en un Sistema Basado en Casos utilizando el agrupamiento conceptual*. 2014.

32. Lazo-Cortés, M., Ruíz-Shulcloper, J. y Alba-Cabrera, E. *Introducción a la teoría de testores*. 1995. 1995.

33. Ruíz-Shulcloper, J., Bravo-Martínez, A. y Águila-Feros, L. Algoritmos BT y TB para el cálculo de todos los test típicos. En: *Revista Ciencias Matemáticas*. 1985, Vol. 6, núm. 2.

34. Lazo-Cortés, M. y Sánchez-Díaz, G. Modificaciones al algoritmo BT para mejorar sus tiempos de ejecución. En: *Revista Ciencias Matemáticas*. 2002, Vol. 20, núm. 2.

35. Guilcapi-Durán, D.F. *Utilización de matrices de pruebas para evaluar el desempeño del cálculo de los testores típicos*. 2012.

36. Alba-Cabrera, E. et al. YYC : A Fast Performance Incremental Algorithm for Finding Typical Testors. En: *Congreso Iberoamericano de Reconocimiento de Patrones*. 2014, p. 416-423.

37. Pons-Porrata, A. y Lias-Rodríguez, A. BR : A New Method for Computing All Typical Testors. En: *Congreso Iberoamericano de Reconocimiento de Patrones*. 2009, p. 433-440.

38. Sánchez-Díaz, G. et al. A Fast Implementation of the CT \_ EXT Algorithm for the Testor Property Identification. En: *Lecture Notes in Computer Science*. 2010, p. 92-103. DOI 10.1007/978-3-642-16773-7.

39. Ruiz-Shulcloper, J. y Carrasco-Ochoa, J.A. Problemas de sensibilidad del conjunto de testores típicos de una matriz booleana. En: *Taller Iberoamericano de Reconocimiento de Patrones. México*. 1998, p. 1-10.

40. Yero-Oses, E.A., Reyes-González, Y. y Martínez-Sánchez, N. CEPAR:Un sistema herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones. En: *Peña Tecnológica*. 2016,

41. Pico-Peña, R. *Prognosis.Sistema Herramienta de reconocimiento de patrones*. 1995.

42. Ruíz-Shulcloper, J. y Pico-Peña, R. SELECTOR: Una herramienta de trabajo para especialistas no matemáticos. En: *Revista CENIC ciencias Biológicas*. 1988, Vol. 10, p. 3.
43. Fuentes-Rodríguez, A. y Ruíz-Shulcloper, J. ACLAPIP: Un sistema de programas para la clasificación basado en el peso de los patrones. En: *Revista Ciencias Matemáticas*. 1986, Vol. 7, núm. 2.
44. Martínez-García, Y. *Aplicación de la técnica de Minería de Datos agrupamiento sobre el área de Gestión Académica de la Universidad de las Ciencias Informáticas*. 2012.
45. Corso, C.L. *Aplicación de algoritmos de clasificación supervisada usando Weka*.
46. Jendrock, E. et al. *Java Platform, Enterprise Edition*. 2014.
47. Mendoza-González, G. *Herramienta de Desarrollo Netbeans*. 2015.
48. Pompa-Torres, L. y Cabrera-González, E.R. Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información Visual Paradigm for UML plug-in for Model Driven Development of Information Management Systems. En: *Serie Científica de la Universidad de las Ciencias Informáticas*. 2012, Vol. 5, núm. 10, p. 1-11.
49. Linstine, H.A. y Turoff, M. *The Delphi Method. Techniques and Applications*. 2002.
50. Landeta, J. *El método Delphi: Una técnica de previsión para la incertidumbre*. 1999.
51. Rivero-Díaz, M. Modelo para la formación integral de los estudiantes de primer año de la Universidad de las Ciencias Informáticas en Cuba. En: *Pedagogía Profesional*. 2012, Vol. 10, núm. 2.
52. Hernández-Leonard, A.R. Evaluación de la satisfacción con el servicio de capacitación del INIMET. En: *Boletín Científico Técnico INIMET*. 2013, núm. 1.

- ALBA-CABRERA, E., GUILCAPI, D. y IBARRA-FIALLO, J., 2012. New Strategies for Evaluating the Performance of Typical Testor Algorithms. *Congreso Iberoamericano de Reconocimiento de Patrones*, pp. 813-820.
- ALBA-CABRERA, E., IBARRA-FIALLO, J. y GODOY-CALDERÓN, S., 2013. A Theoretical and Practical Framework for Assessing the Computational Behavior of Typical Testor-Finding Algorithms. *Congreso Iberoamericano de Reconocimiento de Patrones*, pp. 351-358.
- ALBA-CABRERA, E., IBARRA-FIALLO, J., GODOY-CALDERÓN, S. y CERVANTES-ALONSO, F., 2014. YYC : A Fast Performance Incremental Algorithm for Finding Typical Testors. *Congreso Iberoamericano de Reconocimiento de Patrones*, pp. 416-423.
- CABALLERO-MOTA, Y., ALVAREZ, D., BALTA, A., BELLO-PÉREZ, R.E. y GARCÍA-LORENZO, M.M., 2007. Un nuevo algoritmo de selección de rasgos basado en la teoría de los conjuntos aproximados. *Revista Facultad de Ingeniería Unoversidad de Antioquia Colombia*, no. 41, pp. 132-144.
- CASTAÑEDA-CAO, O. y GARCÍA-LORENZO, M.M., 2015. Análisis y propuesta de selección de rasgos para el Reconocimiento de Expresiones Faciales. *Revista Cubana de Ciencias Informáticas*, vol. 9, no. 1, pp. 45-62.
- COELLO-GONZÁLEZ, S. y HERNÁNDEZ-LEÓN, R.A., 2002. *El Paradigma Cuantitativo de la Investigación científica*. S.l.: s.n. ISBN 9591603436.
- CORSO, C.L., [sin fecha]. *Aplicación de algoritmos de clasificación supervisada usando Weka*. S.l.: s.n.
- ESPINOSA-ALBA, I. y GUTIÉRREZ-ROSA, L.T., 2010. *La minería de datos como soporte a la toma de decisiones estratégicas de las organizaciones*. S.l.: s.n.
- FUENTES-RODRÍGUEZ, A. y RUÍZ-SHULCLOPER, J., 1986. ACLAPIP : Un sistema de programas para la clasificación basado en el peso de los patrones. *Revista Ciencias Matemáticas*, vol. 7, no. 2.
- GÓMEZ-HERRERA, J.E., RODRÍGUEZ-MORÁN, O., VALLADARES-AMARO, S., RUIZ-SHULCLOPER, J., PICO-PEÑA, R., ECHEVARRÍA-RODRIGUEZ, G., TENREYRO-PÉREZ, R., OTERO-MARRERO, R., CHEREMESINA, E., CRUZ-TOLEDO, R., BARCELÓ-CAROL, G., ALVAREZ-CASTRO, J., BAREA-CENTENO, M. y GARCÍA-SÁNCHEZ, R., 1994. Pronóstico gasopetrolífero en la asociación ofiolítica cubana aplicando modelación matemática. *Geofísica Internacional*, vol. 33, no. 3, pp. 447-467.
- GONZÁLEZ-GUEVARA, V.I., GODOY-CALDERÓN, S., ALBA-CABRERA, E. y IBARRA-FIALLO, J., 2015. A Mixed Learning Strategy for Finding Typical Testors in Large Datasets. *Congreso Iberoamericano de Reconocimiento de Patrones*, pp. 716-723. DOI 10.1007/978-3-319-25751-8.

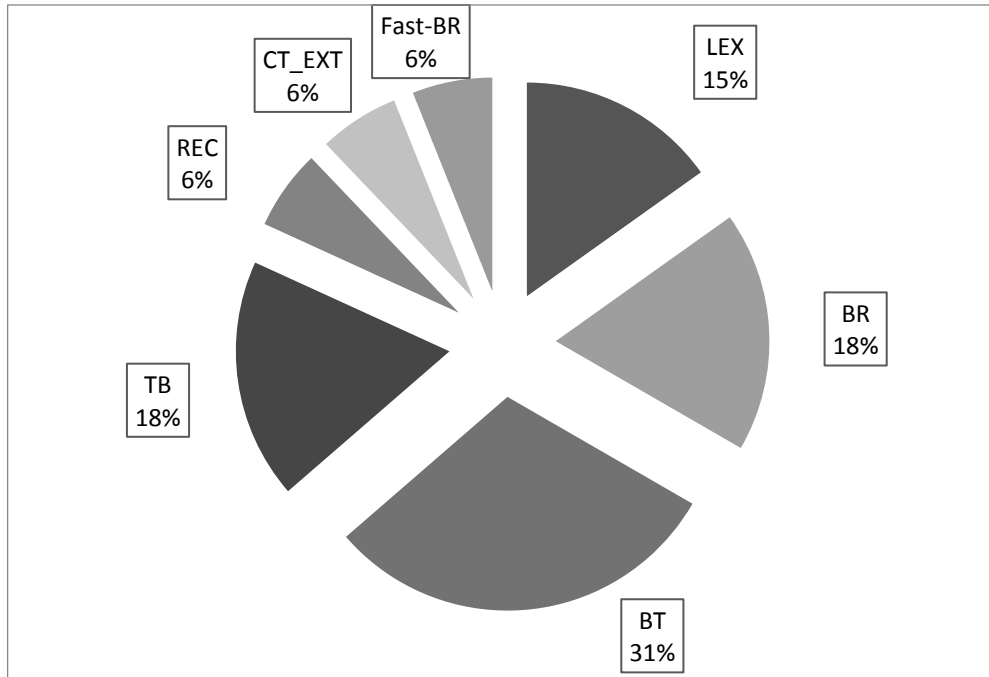
- GUERRA-GANDÓN, A., VEGA-PONS, S. y RUIZ-SHULCLOPER, J., 2012. Algoritmos de agrupamiento conceptuales: un estado del arte. *Serie Azul*, CENATAV,
- GUILCAPI-DURÁN, D.F., 2012. *Utilización de matrices de pruebas para evaluar el desempeño del cálculo de los testores típicos*. S.l.: s.n.
- GUZMÁN-ARENAS, A. y MARTÍNEZ-TRINIDAD, J.F., 2001. The logical combinatorial approach to pattern recognition, an overview through selected works. *Pattern Recognition*, vol. 34, pp. 741-751.
- HERNÁNDEZ-LEONARD, A.R., 2013. Evaluación de la satisfacción con el servicio de capacitación del INIMET. *Boletín Científico Técnico INIMET*, no. 1.
- JENDROCK, E., CERNERA-NAVARRO, R., EVANS, I., HAASE, K. y MARKITO, W., 2014. *Java Platform, Enterprise Edition*. S.l.: s.n.
- LANDETA, J., 1999. *El método Delphi: Una técnica de previsión para la incertidumbre*. S.l.: s.n.
- LAZO-CORTÉS, M., RUÍZ-SHULCLOPER, J. y ALBA-CABRERA, E., 2001. An overview of the evolution of the concept of testor. *Pattern Recognition*, vol. 34, no. 4, pp. 753-762. ISSN 00313203. DOI 10.1016/S0031-3203(00)00028-5.
- LAZO-CORTÉS, M., RUÍZ-SHULCLOPER, J. y ALBA-CABRERA, E., 1995. *Introducción a la teoría de testores*. 1995. S.l.: s.n.
- LAZO-CORTÉS, M. y SÁNCHEZ-DÍAZ, G., 2002. Modificaciones al algoritmo BT para mejorar sus tiempos de ejecución. *Revista Ciencias Matemáticas*, vol. 20, no. 2.
- LINSTINE, H.A. y TUROFF, M., 2002. *The Delphi Method. Techniques and Applications*. S.l.: s.n.
- MARÍA-SEIJAS, L., 2011. *Reconocimiento de patrones utilizando técnicas estadísticas y conexionistas aplicadas a la clasificación de dígitos manuscritos*. S.l.: s.n.
- MARTÍNEZ-GARCÍA, Y., 2012. *Aplicación de la técnica de Minería de Datos agrupamiento sobre el área de Gestión Académica de la Universidad de las Ciencias Informáticas*. S.l.: s.n.
- MARTÍNEZ-SÁNCHEZ, N. y REYES-GONZÁLEZ, Y., 2014. *Algoritmos Conceptuales : Una perspectiva para el modelado del estudiante en los Sistemas Tutoriales Inteligentes*. S.l.: s.n.
- MENDOZA-GONZÁLEZ, G., 2015. *Herramienta de Desarrollo Netbeans*. S.l.: s.n.
- ORTÍZ-POSADAS, M.R., MARTÍNEZ-TRINIDAD, J.F. y RUÍZ-SHULCLOPER, J., 1996. A new approach to differential diagnosis of diseases. *Bio-Medical Computing*, vol. 40, pp. 179-185. ISSN 00207101. DOI 10.1016/0020-7101(95)01143-9.
- ORTÍZ-POSADAS, M.R., MAYA-BEHAR, J. y LAZO-CORTÉS, M., 1998. Evaluación de la cirugía de labio y paladar hendido con el enfoque lógico combinatorio de la teoría de reconocimiento de patrones. *Cuaderno de Ingeniería Biomédica*, vol. 14, pp. 7-22.



- ORTÍZ-POSADAS, M.R. y VEGA-ALVARADO, L., 2001. *Análisis de una muestra de pacientes con labio- paladar hendido usando un algoritmo de tipicidad y contraste*. S.l.: s.n.
- ORTÍZ-POSADAS, M.R., VEGA-ALVARADO, L., JIMÉNEZ-JACINTO, V. y LAZO-CORTÉS, M., [sin fecha]. *El concepto de analogía en medicina. Una función de semejanza para pacientes con fisuras de paladar*. S.l.: s.n.
- PICO-PEÑA, R., 1995. *Prognosis.Sistema Herramienta de reconocimiento de patrones*. S.l.: s.n.
- PICO-PEÑA, R., ÁLVAREZ-GÓMEZ, L. y COTILLA-RODRÍGUEZ, M., [sin fecha]. *Zonación sísmica de la isla de Cuba mediante algoritmos de clasificación lógico combinatorio*. S.l.: s.n.
- POMPA-TORRES, L. y CABRERA-GONZÁLEZ, E.R., 2012. Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información Visual Paradigm for UML plug-in for Model Driven Development of Information Management Systems. *Serie Científica de la Universidad de las Ciencias Informáticas*, vol. 5, no. 10, pp. 1-11.
- PONS-PORRATA, A. y LIAS-RODRÍGUEZ, A., 2009. *Un nuevo algoritmo de escala exterior para el cálculo de los testores típicos*. S.l.: s.n.
- PONS-PORRATA, A. y LIAS-RODRÍGUEZ, A., 2009. BR : A New Method for Computing All Typical Testors. *Congreso Iberoamericano de Reconocimiento de Patrones*, pp. 433-440.
- PONS-PORRATA, A., RUIZ-SHULCLOPER, J. y BERLANGA-LLAVORI, R., 2003. *A Method for the Automatic Summarization of Topic-Based Clusters of Documents*. 2003. S.l.: s.n.
- PONS-PORRATA, A. y SANTIESTEBAN-ALGANZA, Y., 2003. LEX: Un nuevo algoritmo para el cálculo de los testores típicos. *Revista Ciencias Matemáticas*, vol. 21, no. 1.
- REYES-GONZÁLEZ, Y., 2014. *Modelo para la adaptación de las soluciones en un Sistema Basado en Casos utilizando el agrupamiento conceptual*. S.l.: s.n.
- RIVERO-DÍAZ, M., 2012. Modelo para la formación integral de los estudiantes de primer año de la Universidad de las Ciencias Informáticas en Cuba. *Pedagogía Profesional*, vol. 10, no. 2.
- RODRÍGUEZ-DIEZ, V., MARTÍNEZ-TRINIDAD, J.F., CARRASCO-OCHOA, J.A., LAZO-CORTÉS, M., FERREGRINO-URIBE, C. y CUMPLIDO, R., 2015. A fast hardware software platform for computing irreducible testors. *Expert Systems With Applications*, vol. 42, pp. 9612-9619. DOI 10.1016/j.eswa.2015.07.037.
- ROJAS-DELGADO, J., 2016. Selección de rasgos en muestras citológicas usando información heurística. *Revista Cubana de Ciencias Informáticas*, vol. 10, no. 2, pp. 73-88.
- RUÍZ-SHULCLOPER, J., 2009. *Reconocimiento Lógico Combinatorio De Patrones: Teoría Y Aplicaciones*. S.l.: s.n.
- RUÍZ-SHULCLOPER, J., 2013. Acerca del surgimiento del Reconocimiento de Patrones en Cuba. *Revista Cubana de Ciencias Infórmaticas*, vol. 7, no. 2. ISSN 2227-1899.

- RUÍZ-SHULCLOPER, J., BRAVO-MARTÍNEZ, A. y ÁGUILA-FEROS, L., 1985. Algoritmos BT y TB para el cálculo de todos los test típicos. *Revista Ciencias Matemáticas*, vol. 6, no. 2.
- RUIZ-SHULCLOPER, J. y CARRASCO-OCHOA, J.A., 1998. Problemas de sensibilidad del conjunto de testores típicos de una matriz booleana. *Taller Iberoamericano de Reconocimiento de Patrones. México*, pp. 1-10.
- RUÍZ-SHULCLOPER, J., CARRASCO-OCHOA, J.A. y MARTÍNEZ-TRINIDAD, J.F., 2013. Reconocimiento de patrones: conceptos y metodología. *Serie Azul, CENATAV*, ISSN 1098-6596. DOI 10.1017/CBO9781107415324.004.
- RUIZ-SHULCLOPER, J. y FUENTES-RODRÍGUEZ, J., 1981. Modelo cibernético para el análisis de la delincuencia juvenil. *Ciencias matemáticas*, vol. 2, no. 1, pp. 141-153. ISSN 0256-5374.
- RUÍZ-SHULCLOPER, J. y PICO-PENÑA, R., 1988. SELECTOR: Una herramienta de trabajo para especialistas no matemáticos. *Revista CENIC ciencias Biológicas*, vol. 10, pp. 3.
- SANCHEZ-DIAZ, G., LAZO-CORTÉS, M.S. y FUENTES-CHÁVEZ, O., 1999. Algoritmo genético para calcular de costo mínimo testores típicos. *Simposio Iberoamericano de Reconocimiento de Patrones*,
- SÁNCHEZ-DÍAZ, G., PIZA-DÁVILA, I., LAZO-CORTES, M., MORA-GONZÁLEZ, M. y SALINAS-LUNA, J., 2010. A Fast Implementation of the CT \_ EXT Algorithm for the Testor Property Identification. *Lecture Notes in Computer Science.*, pp. 92-103. DOI 10.1007/978-3-642-16773-7.
- TOBERGTE, D.R. y CURTIS, S., [sin fecha]. *Aplicación de técnicas de reconocimiento de patrones con enfoque lógico combinatorio en la clasificación de un complejo de géneros de hongos patógenos en la caña de azúcar*. S.l.: s.n. ISBN 9788578110796.
- TORRES-SOTO, M.D., TORRES-SOTO, A., CUELLAR-GUZMÁN, F., TORRES-SOTO, M. de la L. y PONCE-DE-LEÓN-SENTÍ, E.E., 2013. *Testores en la determinación de factores de riesgo: TRALI*. S.l.: s.n.
- YERO-OSES, E.A., REYES-GONZÁLEZ, Y. y MARTÍNEZ-SÁNCHEZ, N., 2016. CEPAR: Un sistema herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones. *Peña Tecnológica*,

**Anexo 1:** Partiendo de la encuesta realizada los algoritmos que no deben faltar para el cálculo de todos los TTs según el criterio de los usuarios son los siguientes:



**Figura 16. Porcentaje de algoritmos**

**Anexo 2:** Encuesta Delphi aplicada a los expertos.

*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: *Extensión de la herramienta CEPAR con algoritmos de escala exterior para el cálculo de Testores Típicos*. 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,*

*Eyidey Pérez Hernández y Paloma García Figarola.*

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 3:** Encuesta ladov aplicada a usuarios potenciales de la propuesta.

*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: *Extensión de la herramienta CEPAR con algoritmos de escala exterior para el cálculo de Testores Típicos*. 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,*

*Eyidey Pérez Hernández y Paloma García Figarola.*

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 exterior para el cálculo de todos los Testores Típicos (TTs).

Si  No  Me da lo mismo

**8.** ¿Qué algoritmos de escala exterior para el cálculo de todos los TTs 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?