

*UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS*

*FACULTAD 2*



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

*Título: “Herramienta computacional para el análisis del  
comportamiento de los estudiantes en la selección de asignaturas  
optativas empleando Reconocimiento Lógico Combinatorio de  
Patrones”*

***Autores:***

*Yisel María Pavón Hernández*

*Mayara Pájaro Castillo*

***Tutores:***

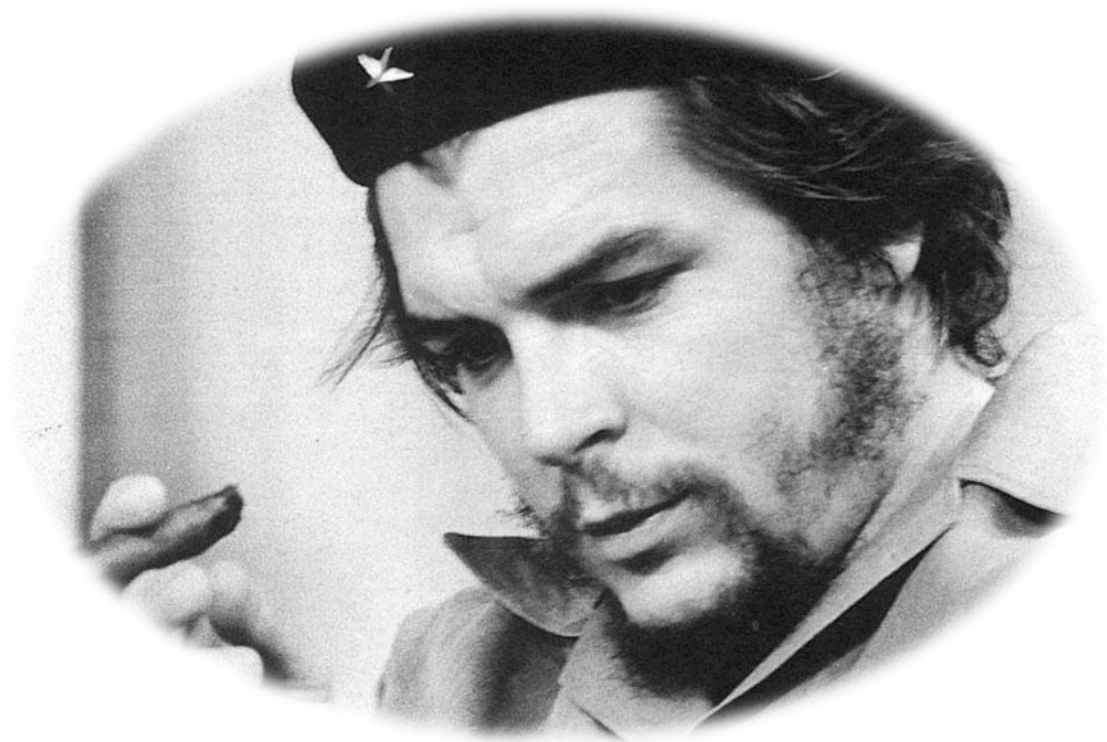
*Msc. Yunia Reyes González*

*Dra. Natalia Martínez Sánchez*

***Co-tutor:***

*Ing. Ernesto Alejandro Yero Oses*

20 de junio de 2017  
“Año 59 de la Revolución”



*“No se vive celebrando victorias, sino superando derrotas”*

**DECLARACIÓN DE AUTORÍA**

Declaramos ser los únicos autores de este trabajo “Herramienta computacional para el análisis del comportamiento de los estudiantes en la selección de asignaturas optativas empleando Reconocimiento Lógico Combinatorio de Patrones”.

Autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, firmamos la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Autores**

\_\_\_\_\_  
**Yisel María Pavón Hernández**  
**Firma Autor**

\_\_\_\_\_  
**Mayara Pájaro Castillo**  
**Firma Autor**

**Tutores**

\_\_\_\_\_  
**Dra. Natalia Martínez Sánchez**  
**Firma Tutor**

\_\_\_\_\_  
**Msc. Yunia Reyes González**  
**Firma Tutor**

**Co-Tutor**

\_\_\_\_\_  
**Ing. Ernesto Alejandro Yero Oses**  
**Firma Co-Tutor**

## Datos de Contacto

**Autor:**

Yisel María Pavón Hernández

Universidad de las Ciencias Informáticas, La Habana, Cuba.

**Email:** [ympavon@estudiantes.uci.cu](mailto:ympavon@estudiantes.uci.cu)

**Autor:**

Mayara Pájaro Castillo

Universidad de las Ciencias Informáticas, La Habana, Cuba.

**Email:** [mpajaro@estudiantes.uci.cu](mailto:mpajaro@estudiantes.uci.cu)

**Tutor:**

Dra. Natalia Martínez Sánchez

Universidad de las Ciencias Informáticas, La Habana, Cuba.

**Email:** [natalia@uci.cu](mailto:natalia@uci.cu)

**Tutor:**

Msc. Yunia Reyes González

Universidad de las Ciencias Informáticas, La Habana, Cuba.

**Email:** [yrglez@uci.cu](mailto:yrglez@uci.cu)

**Co-Tutor:**

Ing. Ernesto Alejandro Yero Oses

Universidad de las Ciencias Informáticas, La Habana, Cuba.

**Email:** [yero@uci.cu](mailto:yero@uci.cu)

## **AGRADECIMIENTOS**

### **Agradecimientos generales**

A los tutores, Natalia, Yunia y Yero por enseñarnos los conocimientos básicos necesarios para poder enfrentar el desarrollo de la investigación.

### **Agradecimientos de Yisel**

A mi familia y a todas las personas que he conocido a lo largo de mis pasos por la universidad y que de una forma u otra han formado parte de mi vida. .

### **Agradecimientos de Mayara**

A mis padres pero sobre todo a mi mamá por toda su paciencia, por enseñarme que todo con perseverancia se logra, por aconsejarme, por permitirme lograr este sueño y por ser la mejor madre del mundo, a ella le agradezco el estar hoy donde estoy y ser la persona que soy.

A mis hermanos ya que gracias ellos me he esforzado tanto para ser un ejemplo para sus vidas que recién comienzan.

A todos mis amigos (la familia que uno escoge), todos los que he conocido a lo largo de este paso por la universidad, les agradezco por los mejores años de mi vida, años que nunca olvidare pues pasamos por muchas cosas juntos. En especial a mi miji Elaine y a mi loquita Yisel (mi compañera de tesis) por ser las hermanas que nunca tuve, aunque siempre estemos discutiendo y muchas veces no estemos de acuerdo en todo, nos aconsejamos y nos ayudamos mutuamente; también a Rosi, Elizabet, Yilian(Pelusa), Lisvet, Yilian(Flaca) las que siempre me han soportado incluso en mis peor día, aunque yo también las he tenido que soportar a ellas jajaja. A mis amigos masculinos, Albertico, Erik, Manuel, Aroldo, Yordi, Javier, Jimmy, Adrian, Miguelito, a mis negros Lionys y Dariel. A los que ya no están conmigo pero también son amigos, en general a todos con los que de una forma u otra he compartido en algún momento.

A mi familia en general por su ayuda, su apoyo incondicional y por estar siempre pendientes de mis estudios en especial mi tía Elisenda.

A mi otra familia, la familia de mi hermano, ellos que me acogieron como otra hija más; Zaida, Nena, Levisnay, la gente de Guanabo, a Corvays por darme a la personita más importante de mi vida Duvarys y ser como un padre para mí.

A todos los profesores que he conocido y me han ayudado en el paso por la universidad en especial a mi profesor Omar gracias por todo.

## Dedicatoria

### DEDICATORIA

#### *Dedicatoria de Yisel*

*A mi papá, a mi mamá, a mi hermana, a mi cuñado, a mi abuela nena y a mi novio por ser las personas que me impulsan cada día a seguir adelante.*

#### *Dedicatoria de Mayara*

*A mi Mamá por ser mi todo.*

*A todas aquellas personas que han contribuido de una forma u otra con mi formación profesional, a mis tutores, mis compañeros, mis profesores y mis amigos.*

### RESUMEN

La formación integral de los estudiantes universitarios en Cuba debe dar como resultado graduados con un sólido desarrollo político desde los fundamentos de la Ideología de la Revolución Cubana; dotados de una amplia cultura científica, ética, jurídica, humanista, económica, medio ambiental y competentes para el desempeño profesional. Por tal motivo los planes de estudios están en constante perfeccionamiento.

En este trabajo se presenta una herramienta computacional de apoyo a la toma de decisiones como parte del perfeccionamiento del currículo optativo de la carrera de Ingeniería en Ciencias Informáticas utilizando un modelo híbrido que integra el Razonamiento Basado en Casos y el Reconocimiento Lógico Combinatorio de Patrones. Además, se presenta la validación experimental del software.

**Palabras Clave:** Herramienta Computacional, Razonamiento Basado en Casos, Razonamiento Lógico Combinatorio de Patrones.

ÍNDICE	
INTRODUCCIÓN .....	1
CAPÍTULO I. REFERENTES TEÓRICOS.....	8
1.1 Ingeniería del Conocimiento .....	8
1.2 Razonamiento Basado en Casos (10) .....	10
1.2.1 Modelo de la Base de Casos .....	13
1.2.2 Selección de rasgos en los SBC. ....	14
1.3 El enfoque lógico combinatorio en la Clasificación No Supervisada.....	15
1.3.1 Algoritmos de Agrupamiento Conceptual .....	16
1.4 Análisis de soluciones existentes .....	17
1.4.1 Sistema de Razonamiento Basado en Casos para el Soporte a la Toma de Decisiones (19) .....	18
1.4.2 Técnicas de agrupamiento aplicadas a los registros de navegación por internet (20).....	18
1.4.3 Herramienta WEKA (Plataforma para la experimentación del Aprendizaje Automático) .....	19
1.4.4 Herramienta CEPAR (Plataforma Entorno Cubano de Reconocimiento de Patrones).....	19
1.7 Análisis crítico de soluciones existentes .....	20
1.8 Metodología de desarrollo .....	21
1.8.1 Metodología de desarrollo seleccionada: Extreme Programming (XP) .....	22
1.9 Tecnologías y herramientas.....	23
1.9.1 Lenguaje de programación.....	23
1.9.2 Herramientas.....	24
1.10 Consideraciones Parciales .....	25
CAPÍTULO II. MODELO HÍBRIDO BASADO EN EL HOLOTIPO DE LA CLASE PARA LA TOMA DE DECISIONES EN UN SBC. ....	27
2.1 Descripción del modelo propuesto.....	27
2.2 Metodología para el desarrollo del software HCAO.....	33
2.2.1 Fase de Exploración.....	33
2.2.2 Fase de Planificación .....	37
2.2.3 Fase de Diseño .....	39
2.3 Arquitectura del software .....	40
2.4 Patrones de Diseño .....	41
2.5 Patrón Arquitectónico.....	42
2.6 Consideraciones Parciales .....	42



CAPÍTULO III. Pruebas e interpretación de los resultados. ....	44
3.1 Descripción de la aplicación realizada para el análisis del comportamiento de los estudiantes en la selección de asignaturas optativas.....	44
3.2 Análisis e interpretación de los resultados.....	47
3.3 Características generales de la herramienta HCAO .....	48
3.4 Fase de Prueba .....	50
3.5 Consideraciones Parciales .....	52
CONCLUSIONES GENERALES.....	53
RECOMENDACIONES .....	54
REFERENCIAS.....	55
Anexos .....	60
Anexo1. Historia de Usuario .....	60

### INTRODUCCIÓN

La universidad es por excelencia la institución social con mayor capacidad para preservar, desarrollar y difundir la cultura en su sentido más amplio, luego es de esperar que ponga el conocimiento más avanzado al servicio y salvaguarda de la humanidad, de la manera más integral e inclusiva posible.

En la actualidad, la educación superior cubana dirige sus acciones en el mantenimiento de un modelo de universidad moderna, humanista, universalizada, científica, tecnológica, innovadora, integrada a la sociedad y profundamente comprometida con la construcción de un socialismo próspero y sostenible. Una universidad caracterizada por la formación de valores y por el aseguramiento de la calidad de sus procesos sustantivos, en aras de lograr un egresado que posea cualidades personales, cultura y habilidades profesionales que le permitan desempeñarse con responsabilidad social, y que propicie su educación para toda la vida. Para el logro de lo anterior, uno de los retos a vencer es contar con diseños curriculares pertinentes que sienten las bases para propiciar un incremento continuo de la calidad y la pertinencia en la formación integral de los profesionales del país<sup>1</sup>.

En la Universidad de las Ciencias Informáticas (UCI) se estudia la carrera de Ingeniería en Ciencias Informáticas (ICI) cuyo Plan de Estudios D<sup>2</sup> está en constante perfeccionamiento para contribuir con eficacia a la misión de la universidad de formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática y desarrollar aplicaciones y servicios

---

<sup>1</sup> Documento Base para el diseño de los planes de estudio "E" Ministerio de Educación Superior de Cuba

<sup>2</sup> Documento oficial del Plan de Estudios D de la carrera de Ingeniería en Ciencias Informáticas. UCI, 2014.

## Introducción

informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática.

La flexibilidad curricular se debe manifestar fundamentalmente por la existencia de tres tipos de contenidos curriculares (base, propio y optativo/electivo), y permitan la actualización permanente del plan de estudio de la carrera y su adaptación a las necesidades del país, del territorio, al desarrollo del claustro y a los intereses de los estudiantes.

Los contenidos curriculares optativos/electivos se ofertan por cada carrera, y son elegidos por los estudiantes. Sirven como complemento para su formación integral, con lo cual se da respuesta también a legítimos intereses de desarrollo personal de cada estudiante.

En la UCI se implementó un proceso de oferta de más de 20 asignaturas optativas en los dos semestres del año académico de cinco de las 14 disciplinas que conforman el Plan de Estudio. Los estudiantes del tercer, cuarto y quinto año académico solicitan cinco de las asignaturas en orden de prioridad para que se les asignen dos según lo establecido en dicho plan.

Durante los últimos tres cursos académicos se está almacenando toda la información correspondiente a este proceso y es de interés del colectivo de la carrera realizar un análisis cualitativo de esta información que permita adoptar decisiones en aras de perfeccionar el currículo optativo.

Los sistemas basados en el conocimiento (1) constituyen técnicas de la Inteligencia Artificial (2) válidas para enfrentar la construcción de sistemas que apoyen la adopción de decisiones dado por sus aspectos afines. Una característica distintiva de los sistemas basados en el conocimiento es la separación del conocimiento (base de conocimiento) del método de solución

## Introducción

del problema (máquina de inferencia). La construcción de la base del conocimiento lleva implícito un arduo proceso de adquisición del conocimiento y es particular para cada sistema, por lo que será necesario construirla para cada aplicación. Sin embargo, la máquina de inferencia puede reusarse en la construcción de varios sistemas basados en el conocimiento siempre que el tipo de conocimiento y el tipo del razonamiento sea similar.

Diferentes tipos de conocimiento dan lugar a diferentes tipos de sistemas basados en el conocimiento, entre ellos los sistemas basados en reglas, los sistemas basados en probabilidades, sistemas expertos conexionistas o redes expertas y los sistemas basados en casos (3).

La máquina de inferencia es el método implementado que utiliza el conocimiento de la base para resolver los problemas del dominio. El tipo de conocimiento determina qué método de solución de problemas es posible utilizar.

En particular resulta claro que para el desarrollo de un sistema de apoyo a la toma de decisiones en el dominio de aplicación planteado se requiere de algoritmos de clasificación no supervisada (4) para facilitar el estudio del comportamiento de los objetos que conforman la base de datos con la cual se trabaja.

Por otra parte, el Reconocimiento Lógico Combinatorio de Patrones (RLCP) (4) proporciona un marco teórico para el estudio y solución de este tipo de problemas, y su esencia consiste en trabajar con datos numéricos y no numéricos de manera simultánea en los objetos, considerando además la ausencia de información.

Por problemas de reconocimiento de patrones se entienden todos aquellos relacionados con la clasificación de objetos y fenómenos y con la determinación de

los factores que inciden en los mismos (5) . Por otro lado, en muchas ciencias aplicadas está presente el problema de revelar la estructura subyacente en una colección de objetos.

En este trabajo se selecciona el RLCP dando continuidad a los resultados de las investigaciones científicas del grupo de Inteligencia Artificial y Reconocimiento de Patrones de la UCI (6) (7) (8) en problemas similares.

En términos de preguntas de investigación el **problema a resolver** se formula entonces así ¿Cómo analizar la información existente sobre el comportamiento de los estudiantes en la selección de las asignaturas del currículo optativo del plan de estudios?

Por tanto, la presente investigación tiene como **objeto de estudio** los algoritmos de clasificación no supervisada del RLCP.

Enmarcado en el **campo de acción** proceso de selección de asignaturas optativas aplicando Razonamiento Basado en Casos y los algoritmos del RLCP.

Para orientar la labor investigativa, la pregunta general que antes se formuló se desglosa en las siguientes **interrogantes científicas**:

1. ¿Qué facilidades ofrece el Razonamiento Basado en Casos para implementar un sistema de esta índole?
2. ¿Cómo diseñar un caso para que el sistema pueda realizar sus funciones?
3. ¿Cómo garantizar un diagnóstico, cómo orientar a un nuevo estudiante utilizando el Razonamiento Basado en Casos?

El **objetivo general** del presente trabajo puede plantearse entonces de la manera siguiente: Diseñar una herramienta computacional que integre el

Razonamiento Basado en Casos y los algoritmos del RLCP de apoyo a la toma de decisiones en el proceso de oferta y asignación de las asignaturas optativas del Plan de estudio de la carrera de ICI.

Para lograr este objetivo general se plantean los siguientes **objetivos específicos**:

1. Elaborar el marco teórico referencial de la investigación relacionada con el Razonamiento Basado en Caso (RBC) y el RLCP.
2. Formalizar un modelo computacional donde se aplique el Razonamiento Basado en Casos en la consideración del estudiante como entidad principal utilizando algoritmos del RLCP.
3. Realizar la implementación computacional del modelo propuesto.
4. Aplicar el modelo propuesto a la solución del proceso de oferta y solicitud de la Asignaturas Optativas (AO).
5. Realizar las pruebas al sistema para su correcto funcionamiento.

Para el cumplimiento de los objetivos y responder a las preguntas de investigación se trazaron las **tareas de investigación** siguientes:

1. Analizar diferentes tipos de sistemas basados en el conocimiento, en cuanto a la forma de representar el conocimiento, el método de solución de problema utilizado y la complejidad en el proceso de Ingeniería del Conocimiento implícito en cada uno de los mismos para fundamentar la selección del Sistemas Basado en Casos.
2. Determinar una función de semejanza que recupere los casos más semejantes al estudiante para modelar a partir de la información almacenada en los casos.

3. Analizar las herramientas y tecnologías para implementar computacionalmente el modelo propuesto.
4. Realizar pruebas al sistema para garantizar el correcto funcionamiento del software.

Para el desarrollo de las tareas científicas se han combinado diferentes **métodos y procedimientos teóricos y empíricos** (9) de la investigación científica en la búsqueda y procesamiento de la información. Los fundamentales son:

**Métodos generales.** El método sistémico para el desarrollo del sistema computacional y lograr que los elementos que formen parte de la aplicación real sean un todo que funcione de manera armónica; el método histórico lógico y el dialéctico para el estudio crítico de los trabajos anteriores.

**Métodos lógicos.** El método analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta; el método inducción-deducción como vía de la constatación teórica durante el desarrollo de la tesis; el método de modelación para el desarrollo de los algoritmos.

**Métodos empíricos.** El método experimental para comprobar la utilidad de los resultados.

**Métodos matemáticos.** Los métodos de experto y estadísticos para la validación de la guía de orientación a la Ingeniería del Conocimiento, la usabilidad de la herramienta de autor con un enfoque basado en casos.

Después de la revisión de la literatura y el desarrollo consecuente del marco teórico, se formuló la siguiente **Idea a defender**: El Razonamiento Basado en Casos facilita integrar los algoritmos del RLCP de apoyo a la toma de

decisiones en el proceso de oferta y asignación de las asignaturas optativas del plan de estudio de la carrera de ICI.

La tesis está conformada por **tres capítulos**.

En el primer capítulo se realiza una caracterización de los Sistemas Basados en Casos, su desarrollo y tipos existentes. Así como, los algoritmos clásicos del RLCP con énfasis en problemas de clasificación no supervisada. Además, se describe la metodología desarrollo y herramientas utilizadas en esta investigación.

El segundo capítulo aborda se aborda la definición del modelo propuesto utilizando el Razonamiento Basado en Casos, así como los algoritmos que se utilizan de forma complementaria para facilitar la ingeniería del conocimiento.

Además se describen las fases de la metodología de desarrollo seleccionada.

En el tercer capítulo se describe la aplicación del modelo propuesto en la selección de AO y se realiza la validación experimental del sistema computacional desarrollado.

Por último, se establecen las Conclusiones, se emiten Recomendaciones, se relacionan las Referencias Bibliográficas y se incluye un conjunto de Anexos que facilitan la comprensión de la memoria gráfica de la Tesis.



### **CAPÍTULO I. REFERENTES TEÓRICOS**

En este capítulo se hace un estudio referente a los fundamentos teóricos de los Sistemas Basados en Casos y los algoritmos de clasificación no supervisada del RLCP, así como alternativas para la selección de rasgos relevantes y el cálculo de su importancia utilizando los testores típicos. Por último, se hace referencia a la metodología de desarrollo de software necesaria para la implementación de la herramienta computacional que se desarrolla.

#### **1.1 Ingeniería del Conocimiento**

La Ingeniería del Conocimiento (IC) surge como consecuencia de la necesidad de establecer principios metodológicos y científicos que permitan desarrollar sistemas basados en el conocimiento a partir de los fundamentos de la informática en general y de la inteligencia computacional en particular. En este aspecto puede vérsela como la especialización de Ingeniería de Software en su aplicación al desarrollo de Sistemas Inteligentes (3).

La IC se enfoca al desarrollo de sistemas basados en el conocimiento, destacándose la necesidad de la adquisición del conocimiento, así como su especificación, verificación, validación, diseño e implementación en sistemas informáticos o lenguajes apropiados para la construcción de bases de conocimiento para la toma de decisiones.

Para la creación de la base de conocimiento es necesario realizar un arduo proceso de revisión del conocimiento público existente, así como el conocimiento que poseen los expertos en el dominio, conocimiento privado.

El conocimiento público incluye las definiciones, hechos y teorías publicadas, pero la experticidad usualmente incluye más que esta clase de conocimiento.

Los expertos humanos generalmente poseen conocimiento privado. Sobre el

## Capítulo 1 Referentes teóricos

conocimiento público hay consenso, el privado puede llevar a polémicas entre los expertos.

Elucidar y reproducir tal conocimiento es la tarea central en la construcción de sistemas basados en el conocimiento; el sujeto de esta acción es el ingeniero de conocimiento.

La adquisición del conocimiento a partir de expertos humanos, si es necesaria e insustituible en muchas aplicaciones, ha presentado diversas dificultades que van desde la representación del sentido común hasta las excesivas demoras en la implementación y el mantenimiento de los sistemas.

La envergadura del proceso de adquisición del conocimiento depende del tipo de conocimiento (3). En los sistemas basados en reglas se desarrolla un proceso complejo y prolongado pues la extracción se refiere a la formalización de reglas y el pensamiento humano no siempre está regido conscientemente por las reglas de la lógica; en ocasiones es básicamente un procesamiento de información recuperada con el tiempo.

En los sistemas basados en probabilidades la adquisición del conocimiento consiste en coleccionar muestras y realizar un procesamiento estadístico que produzca las probabilidades o frecuencias que forman la base de conocimiento. No son factibles para todo tipo de dominio, pues se dificulta construir las redes con ayuda de expertos humanos cuando existen carencias de conocimiento. No son viables para explicar el razonamiento, ya que los métodos y modelos que utiliza están aún lejos de ofrecer explicaciones comprensibles.

En las redes expertas la adquisición del conocimiento incluye la selección de los ejemplos, el diseño de su topología y el entrenamiento de la red para hallar el conjunto de pesos. Facilitan el trabajo con información incompleta y brindan

## Capítulo 1 Referentes teóricos

algoritmos poderosos de aprendizaje para crear la base de conocimiento; pero requieren de muchos ejemplos y son cajas negras que no explican como la solución se alcanza.

En los sistemas basados en casos la adquisición del conocimiento se reduce a la selección de un conjunto de ejemplos o casos resueltos y su organización en la base de casos. Argumenta una solución mediante los casos que son relevantes al nuevo problema. Cada caso es la experiencia anterior almacenada. Su dificultad radica en la definición adecuada de la función de semejanza, al no existir una función de semejanza general apropiada para cualquier problema.

### **1.2 Razonamiento Basado en Casos (10)**

El Razonamiento Basado en Casos (RBC) (11), es un enfoque que aborda nuevos problemas tomando como referencia problemas similares resueltos en el pasado. De modo que problemas similares tienen soluciones similares, y la similitud juega un rol esencial (12). Sus componentes fundamentales son la base de casos, el módulo de recuperación de casos y el módulo de adaptación de las soluciones.

#### **Base de Casos (BC)**

La BC contiene las experiencias, ejemplos o casos a partir de los cuales el sistema hace sus inferencias. Esta base puede ser generada a partir de casos o ejemplos resultantes del trabajo de expertos humanos o por un procedimiento automático o semiautomático que construye los casos desde datos existentes registrados, por ejemplo, en una base de datos.

Puede ser representada a través de una tabla cuyas columnas son etiquetadas por variables o atributos que representan los rasgos predictores y objetivos; mientras sus filas representan los casos u otras formas de representación.

### **Módulo de Recuperación**

En este módulo se recuperan de la BC los casos más semejantes al problema. No existe una medida de semejanza única, general, para cualquier dominio, de ahí que la eficiencia del sistema radica en la función de semejanza que se defina.

La función de semejanza más sencilla consiste en contar el número de rasgos predictores similares entre ambos, sin embargo, se presenta el problema de que la importancia de los rasgos predictores varía de un contexto a otro (4). La representación más general de función de semejanza pudiera ser:

$$f(O_o, O_t) = \frac{\sum_{i=1}^n w_i \cdot \delta_i(x_i(O_o), x_i(O_t))}{\sum_{i=1}^n w_i}$$

Donde  $n$  es el número de rasgos predictores y  $w_i$  la importancia asociada al rasgo  $i$ .

### **Módulo de Adaptación**

Después de la determinación de los casos más semejantes, las soluciones contenidas en dichos casos pueden usarse directamente como solución al nuevo problema, pero comúnmente necesitan ser modificadas. En las bibliografías (3) y (13) aparecen métodos y reglas de adaptación para realizar dicha modificación.

La forma más simple que puede tomar la adaptación es la adaptación nula que tiene sentido en sistemas de ayuda a la toma de decisiones. En ella la solución

## Capítulo 1 Referentes teóricos

del caso más similar al problema planteado se toma sin cambio alguno. Esto significa que la responsabilidad de hacer los cambios pertinentes queda a cargo del usuario final. Este tipo de adaptación consiste en apoyar las decisiones del usuario presentándole experiencias previas similares al nuevo problema, para que este conforme la solución final atendiendo a estas recomendaciones, su experiencia, y las características del problema.

La adaptación transformacional significa que la solución del caso más similar al problema actual será transformada en la nueva solución del problema. Los elementos que conforman la vieja solución serán reorganizados permitiéndose hacer modificaciones, inserciones y eliminaciones de tales elementos.

Por su parte la adaptación generativa es radicalmente diferente a las anteriores pues requiere de un solucionador de problemas en el dominio específico de la aplicación. Este sistema debe ser capaz de resolver problemas del dominio por si solo (sin la necesidad del RBC) aunque esto implique una gran complejidad computacional.

Una vez propuesta la solución en el proceso de adaptación, esta debe ser evaluada. El proceso de evaluación requiere un elevado conocimiento del dominio, una vía de evaluar la nueva solución es aplicarla y ver qué sucede. Después de este proceso la nueva solución pudiera ser adaptada nuevamente, conocido esto como reparación. Por último, el nuevo problema puede ser incorporado a la base a través del proceso de actualización de la memoria de casos (11).

El enfoque que utilizan los Sistemas Basados en Caso (SBC) para la adquisición de conocimiento es una de las ventajas que se le acreditan a este tipo de sistemas, pues razonan desde episodios específicos lo cual evita el

problema de descomponer el conocimiento del dominio y generalizarlo en reglas.

Otras de las ventajas de los SBC están fundamentadas en la flexibilidad para representar el conocimiento a través de los casos, la organización de la BC y de las estrategias de recuperación y adaptación de los casos y que el usuario puede ser capaz de agregar nuevos casos a la BC sin la intervención experta.

Ventajas lo son también, el rechazo de las soluciones previas al resolver un problema y el almacenar casos que resultó un fracaso, lo que permite advertir sobre problemas potenciales a evitar. Así como también poder fundamentar las soluciones derivadas a partir de casos reales.

Las limitantes de los SBC están en la definición de la función de semejanza y en lo difícil que resulta encontrar una estructura apropiada para describir el contenido de un caso y decidir cómo la memoria de casos debe ser organizada e indexada para un almacenamiento, recuperación y rechazo efectivo.

### **1.2.1 Modelo de la Base de Casos**

Un razonador basado en casos depende de la estructura y el contenido de la base de conocimiento. Hasta el presente, para resolver el problema de organizar una BC, un enfoque ha sido almacenar los casos de forma secuencial y analizarlos todos para resolver el nuevo problema. Este tipo de organización hace lento el proceso de recuperación.

Un método alternativo consiste en particionar los casos en grupos y organizarlos jerárquicamente. Esta jerarquía permite una búsqueda más eficiente ya que se sigue por un determinado camino en dependencia de los valores de los rasgos predictores del nuevo problema.

### 1.2.2 Selección de rasgos en los SBC.

La selección de rasgos es cuestión central tanto en la definición del modelo de la BC como en el modelo de recuperación de casos. Potencialmente, en el conjunto de casos podrían estar todas las propiedades que describen los objetos, pero existen rasgos inútiles que carecen de importancia de acuerdo al dominio de aplicación (14) y (15).

El conjunto de rasgos determina qué información será almacenada en memoria para cada uno de sus elementos, la cual debe permitir la posterior recuperación de los casos semejantes dada la descripción de un nuevo problema. Otro aspecto es que dentro del conjunto de rasgos que se seleccionan no todos tienen la misma importancia y esta diferencia debe tenerse en cuenta para comparar objetos.

En los SBC dentro de los criterios más comúnmente utilizados en el cálculo de la importancia de cada rasgo están: criterio de los especialistas del dominio de aplicación, dispersión de los valores del rasgo, frecuencia del valor dado al rasgo, carácter diferenciante del rasgo, fuerza predictiva del rasgo, entre otros.

En el enfoque lógico combinatorio una alternativa de solución al problema de la selección de rasgos es a partir de la utilización del conjunto de testores típicos.

En esencia, un testor es un conjunto de características (rasgos) que diferencia a elementos (objetos) de clases distintas.

Los testores típicos constituyen variantes minimales de subconjuntos de rasgos, existiendo algoritmos que permiten calcular la importancia del rasgo a partir de éstos (4) y (13).

En la bibliografía (13), (14), (16) y (17) se hace referencia a artículos donde se pueden encontrar ejemplos de algoritmos para el cálculo de los testores típicos

como CC y CT, BT, TB, REC, CER, LEX, BR y FAST BR. Algoritmos están basados en el concepto de testor típico.

### **1.3 El enfoque lógico combinatorio en la Clasificación No Supervisada**

En la bibliografía consultada se aprecia que el enfoque lógico combinatorio puede ser aplicado para resolver problemas de la mayoría de las áreas del saber: reconocimiento de caracteres, diagnóstico médico, teledetección de la tierra, identificación de rostros humanos y huellas digitales, pronóstico de roturas en equipos y maquinarias, análisis de señales e imágenes biomédicas, inspección automática, conteo de células sanguíneas, análisis de los registros de pozos, arqueología, pronóstico de depósito de minerales, análisis de la actividad sísmológica, clasificación de documentos, entre otras.

Una variante para estructurar una base de casos consiste en agrupar los casos a partir de su similitud. Precisamente el agrupamiento es una tarea de la clasificación no supervisada que tiene como objetivo descomponer el conjunto de datos, de forma tal que los objetos que pertenecen al mismo grupo sean tan similares como sea posible y los objetos que pertenecen a grupos diferentes sean tan disimilares como sea posible. En un problema de clasificación sin aprendizaje (también clasificación no supervisada), no se conoce cómo se agrupan los objetos, es justamente el objetivo que se persigue (4) .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), uno de los conceptos más importantes en el Reconocimiento de Patrones.



## Capítulo 1 Referentes teóricos

En un problema de clasificación sin aprendizaje los tres elementos esenciales lo constituyen: El espacio de representación de los objetos, la medida de similaridad ( $\beta$ , función de semejanza) y el criterio de agrupamiento  $\Pi$ , es decir, la manera en que es utilizada la similaridad para la solución del problema planteado. Diferentes algoritmos de agrupamiento fueron desarrollados a partir de estas ideas, que esencialmente forman los grupos sobre la base de objetos muy parecidos o cercanos entre sí, y no tenían en cuenta la utilidad del significado de los agrupamientos obtenidos. Estos algoritmos de agrupamiento, convencionales o tradicionales como también son conocidos en la literatura, presentan importantes limitaciones: 1. Dejan el problema de la interpretación de los grupos al analista de datos. 2. No tienen en cuenta los métodos que los humanos emplean para agrupar objetos.

Las personas tienden a agrupar objetos en categorías caracterizadas por conceptos, en grupos similares teniendo en cuenta algún atributo relevante o más importante que el resto.

Los métodos tradicionales de agrupamiento no toman en consideración ningún concepto o construcciones lingüísticas que las personas usan para describir colecciones de objetos (18). Para enfrentar estas limitaciones a finales de los años 70 e inicios de los 80, Ryszard S. Michalski introdujo un conjunto de ideas que dieron origen al agrupamiento conceptual.

### **1.3.1 Algoritmos de Agrupamiento Conceptual**

El agrupamiento conceptual no solo propone obtener, además de los grupos estructurados a partir de una colección de objetos el significado de esas agrupaciones, sino aportar información sobre el sentido que tiene que los objetos pertenezcan a un mismo grupo, brindar las características o conceptos

## Capítulo 1 Referentes teóricos

para esas agrupaciones, definidas sobre la base de los rasgos en términos de los cuales se describen a esos objetos que conforman los grupos.

La motivación esencial era lograr que el agrupamiento obtenido brindara al especialista del área específica una información acerca del significado conjuntual del agrupamiento.

Los algoritmos clásicos de agrupamiento utilizan el primer procedimiento para formar cada grupo, es decir dan una estructuración extensional de los espacios. Los algoritmos de agrupamiento conceptual se componen de dos tareas fundamentales, las cuales no tienen necesariamente que ser independientes ni realizarse en un orden determinado:

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

### **1.4 Análisis de soluciones existentes**

A partir del estudio del estado del arte relacionado con la temática de la investigación se estudian cuatro de las soluciones existentes con características distintivas a tener en cuenta para el desarrollo del modelo propuesto.

### **1.4.1 Sistema de Razonamiento Basado en Casos para el Soporte a la Toma de Decisiones (19)**

Sistema de RBC diseñado para aconsejar a los estudiantes universitarios sobre qué asignaturas optativas cursar. La finalidad es desarrollar una herramienta de apoyo en la elección de asignaturas optativas, que incorpora información acerca de los conocimientos y preferencias académicas de los estudiantes que han cursado al menos un curso en la universidad. La herramienta utiliza posteriormente esta información para aconsejar a nuevos estudiantes. El rendimiento del sistema obtenido se compara con los resultados proporcionados por una red neuronal artificial de funciones de base radial (son aproximadores universales), que se han utilizado para realizar la comparación porque son algoritmos que modelan problemas a partir de conjuntos de datos, especialmente aplicables a problemas de clasificación donde se manejan datos y conocimiento existente del problema. Englobadas dentro de las redes de aprendizaje supervisado, es muy similar a la arquitectura del perceptrón multi-capas, sin embargo, su forma de aprendizaje es sensiblemente diferente.

### **1.4.2 Técnicas de agrupamiento aplicadas a los registros de navegación por internet (20)**

Proceso de Descubrimiento de Conocimiento en Bases de Datos realizado en el entorno de los registros de navegación por Internet, en donde se combinan técnicas estadísticas, numéricas y de agrupamiento con vista a identificar grupos similares en el uso de las cuotas de navegación por Internet. Obteniéndose grupos (patrones) que describen el uso de las cuotas de navegación por Internet de los diferentes usuarios de la universidad, con vista a

apoyar actividades orientadas a la gestión y seguridad del servicio de navegación por Internet.

### **1.4.3 Herramienta WEKA (Plataforma para la experimentación del Aprendizaje Automático)**

Está constituida 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. 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. Permite declarar nuevos algoritmos y trabaja con ausencia de información, además de ser una herramienta multiplataforma (3), (21) y (22).

### **1.4.4 Herramienta CEPAR (Plataforma Entorno Cubano de Reconocimiento de Patrones)**

CEPAR (por sus siglas en inglés Cuban Environment for Logical Combinatorial Pattern Recognition) 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. Desarrollado, empleando JAVA como lenguaje de programación, dado sus comodidades como lenguaje multiplataforma, CEPAR solo requiere de la Máquina Virtual de Java (por sus siglas en inglés 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 (23).

### **Ventajas:**

- Para todos los tipos de rasgos soporta la ausencia de información.
- La herramienta permite el trabajo con variables cuantitativas y cualitativas (simultáneamente).
- Permite definir criterios de comparación 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.

### **1.7 Análisis crítico de soluciones existentes**

El estudio realizado muestra la existencia de sistemas informáticos inteligentes. Estas aplicaciones se asemejan en cuanto a funcionalidades al sistema que se desea desarrollar, y poseen un conjunto de características que se pueden tener en cuenta para el desarrollo del software, sin embargo no pueden ser utilizadas para la solución de la problemática debido a que:

## Capítulo 1 Referentes teóricos

- No realizan manejo apropiado de los rasgos numéricos y no numéricos.
- No tienen implementado los algoritmos clásicos del RLCP para la selección de rasgos, el cálculo de su importancia así como otros algoritmos de clasificación no supervisada.
- Son específica para el dominio de aplicación para la cual fueron desarrolladas.

Solo la herramienta CEPAR está diseñada para soportar el RLCP, de ahí su selección en esta investigación.

### **1.8 Metodología de desarrollo**

Una metodología de desarrollo de software no es más que un conjunto de procedimientos a seguir para estructurar, planificar y controlar el proceso de desarrollo en un sistema, por parte de los desarrolladores (24).

Por una parte están aquellas propuestas más tradicionales, pesadas por el uso exhaustivo de documentación durante todo el ciclo de vida del proyecto, que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas que se usarán, limitando la propia habilidad del equipo para llevar a cabo el proyecto.

Por otra parte, están las metodologías ágiles, que ponen vital importancia en la capacidad de respuesta a los cambios, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un sistema y al desarrollo incremental del software con iteraciones muy cortas, siendo así un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos (24).

Debido a que se necesita estar preparado para cambios durante el proyecto y que se generan pocos artefactos, se adoptó la decisión de utilizar para el desarrollo del sistema una metodología ágil.

### **1.8.1 Metodología de desarrollo seleccionada: Extreme Programming (XP)**

Se selecciona XP como metodología ágil de desarrollo de software ya que pone más énfasis en la adaptabilidad que en la previsibilidad y se adapta a los cambios de requisitos en cualquier punto de vida del sistema. Es una metodología ligera interactiva incremental. XP se basa en una retroalimentación continua entre el cliente y el equipo de desarrollo, se basa en la simplicidad de las soluciones implementadas y acciones para enfrentar los cambios.

Al seleccionarse XP como metodología de desarrollo, el proceso de creación del Sistema se verá guiado por 4 fases, generando en cada una de estas fases varios artefactos (24) (25).

**Fase de Exploración:** en esta fase los clientes plantean a grandes rasgos las historias de usuario que son de interés para la entrega del producto.

**Fase de Planificación:** en esta fase se realiza una estimación del esfuerzo necesario de cada una de las historia de usuario.

**Fase de Diseño:** en esta fase se propone el modelado de las tarjetas clases, Responsabilidad y Colaborador (CRC).

**Fase de Prueba:** en esta fase se comprueba cada una de las funcionalidades implementadas. La metodología XP divide las pruebas del sistema en dos grandes grupos: pruebas unitarias y pruebas de aceptación.

**Se selecciona la metodología ágil de desarrollo XP porque:**

## Capítulo 1 Referentes teóricos

- El equipo de trabajo está compuesto por dos programadores, lo cual disminuye el número de errores durante la programación.
- El grado de interacción entre los miembros del equipo es alto.
- Los requisitos pueden cambiar durante el desarrollo del sistema.
- Cada integrante del equipo de desarrollo debe conocer cualquier parte del código.
- Se cuenta con poco tiempo para el desarrollo del sistema.
- El cliente forma parte del equipo de desarrollo, lo cual ayuda a corregir a tiempo los errores que puedan surgir.

### **1.9 Tecnologías y herramientas**

Las herramientas y tecnologías seleccionadas para desarrollar un sistema deben ser escogidas cuidadosamente, ya que estas podrían suponer el fracaso, por ello resulta vital conocer cuáles son las diferentes alternativas existentes y cuáles son las necesidades específicas del sistema a desarrollar. Luego de analizadas las ideas anteriores en consecuencia con el tipo de proyecto que se desea desarrollar se definieron un conjunto de herramientas y tecnologías válidas para llevar a cabo tal tarea.

#### **1.9.1 Lenguaje de programación**

Un lenguaje de programación es aquel elemento dentro de la Informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis, poniéndose a disposición del programador para que éste pueda comunicarse con los dispositivos de software y hardware. Es un modo práctico para que los seres humanos puedan dar instrucciones a un ordenador o computadora.

- **Java**



## Capítulo 1 Referentes teóricos

El sistema a desarrollar necesita de un lenguaje de programación que permita simplificar el código, posibilitando escribir y utilizar muchas veces un procedimiento, método o hilos de procesos, al igual que su ejecución en varias plataformas. Java permite estas opciones de desarrollo, se pueden heredar y agregar nuevas funcionalidades sin alterar el código inicial. Posee librerías nativas que permiten crear interfaces de usuario con gran fluidez, acortando el tiempo de desarrollo. Por tanto, será utilizado como lenguaje de desarrollo Java en su versión 1.8.

### 1.9.2 Herramientas

#### **Framework de desarrollo**

Un framework de desarrollo o marco de trabajo es una estructura conceptual y tecnológica de soporte definido, además de ser un ambiente de trabajo que contiene librerías de códigos y módulos que pueden ser reutilizados para el rápido desarrollo de aplicaciones.

- **Framework para pruebas unitarias de aplicaciones Java.**

JUnit es un conjunto de clases (framework) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente (26).

- **Herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones (CEPAR).**

## Capítulo 1 Referentes teóricos

CEPAR es una herramienta informática para el reconocimiento lógico combinatorio de patrones, es un sistema herramienta universal, que sirve de apoyo a los docentes, estudiantes e investigadores de este enfoque.

### **IDE de desarrollo**

El Entorno de Desarrollo Integrado (IDE por sus siglas en inglés), es un programa compuesto por un conjunto de herramientas de programación útiles para el desarrollador de software, con el objetivo de mejorar la productividad de este y obtener mayor rapidez en el desarrollo.

#### **– Netbeans**

Netbeans es un entorno integrado de desarrollo gratuito, una herramienta que posee un soporte óptimo para tecnologías Java. Simplifica la gestión de grandes proyectos con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en el trabajo diario. Posee muchas ventajas como agilidad y flexibilidad en la implementación, la creación de aplicaciones multiplataforma, facilidades y garantías para la migración, facilidad de uso, además de ser muy compatible con el lenguaje de desarrollo seleccionado. Por las características antes mencionadas se utilizará el Netbeans en su versión 8.0 (27).

### **1.10 Consideraciones Parciales**

El estudio del estado del arte realizado sirve de fundamento teórico para seleccionar los sistemas basados en caso como un paradigma válido para dar solución al problema planteado. Así como el uso del RLCP como complemento para la selección de los rasgos relevantes, el cálculo de la importancia de éstos, la organización de la base de casos y la implementación de los módulos fundamentales del RBC.

## Capítulo 1 Referentes teóricos

Por otra parte, el estudio realizado de las tecnologías y herramientas informáticas existentes arrojó que la utilización de la metodología ágil XP es pertinente para el desarrollo de la aplicación informática resultante de este trabajo.

Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

**CAPÍTULO II. MODELO HÍBRIDO BASADO EN EL HOLOTIPO DE LA CLASE PARA LA TOMA DE DECISIONES EN UN SBC.**

En este capítulo se conceptualiza un modelo computacional para la elaboración de los componentes fundamentales de los SBC, integrando el paradigma del Razonamiento Basado en caso y el RLCP. Se describen los algoritmos que conforman el modelo; así como los algoritmos que se utilizan de forma complementaria para facilitar la ingeniería del conocimiento en la definición y la organización de la Base de Caso en una estructura jerárquica. Además, se exponen los artefactos generados propios de la metodología de desarrollo utilizada (metodología XP).

**2.1 Descripción del modelo propuesto**

La figura 1 ilustra la idea general del SBC elaborado utilizando algoritmos básicos del RLCP, tomando como fundamento teórico el holotipo de la clase.

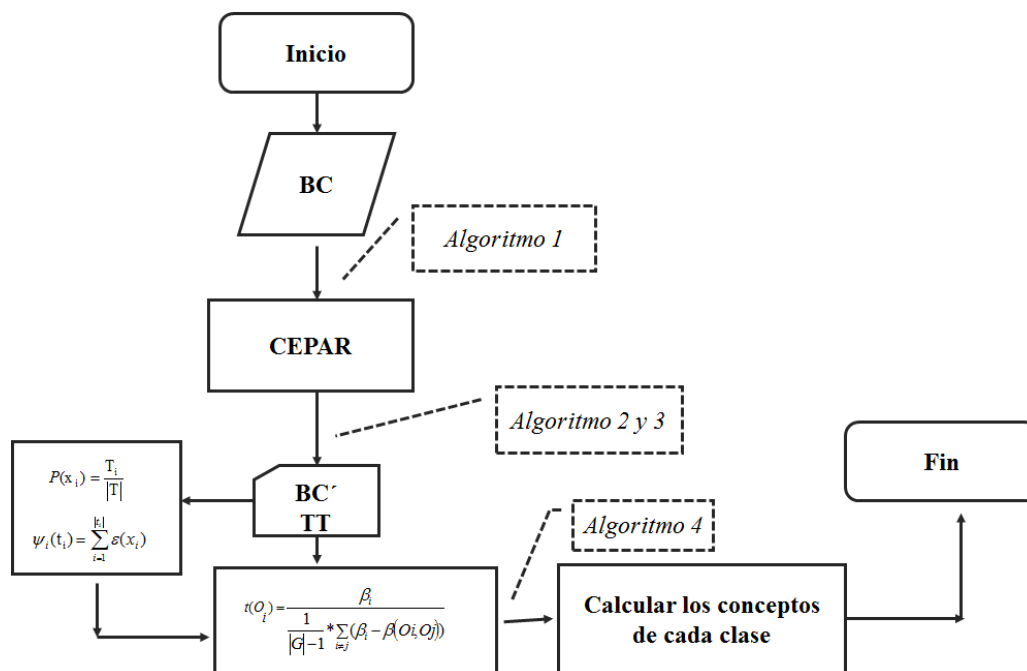


Figura 1. Esquema general del modelo computacional

## Capítulo II Modelo híbrido basado en el holotipo de la clase para

### la toma de decisiones en un SBC

Aplicando el algoritmo 1 se estructura la BC en clases a partir del grado de semejanzas entre los casos y teniendo en cuenta la importancia asignada a cada rasgo según el criterio de los expertos.

---

**Algoritmo 1.** Organización de la BC en clases.

**Entrada:** BC // se corresponde con la MI  
 $W=\{w_1, w_2, \dots, w_r\}$  // conjunto de la importancia  
de  
cada rasgo predictor

**Salida:** BC' //BC dividida en clases

**Paso1.** Se utiliza la herramienta computacional  
CEPAR

---

Con la estructuración de la BC en clases, aún no se logra cumplir con el objetivo general de este trabajo, pues se forman los grupos de estudiantes a partir del grado de semejanza que existe entre ellos según los valores que toman los rasgos predictores definidos, pero no se les ofrece a los expertos las características que distinguen cada grupo, propiedad que les permite analizar y adoptar decisiones en el perfeccionamiento de este proceso.

En el enfoque lógico combinatorio existen diferentes algoritmos que pueden ser utilizados para determinar estas características ya sea calculando los conceptos o determinando un objeto que represente el grupo: algoritmo para el cálculo del ideal de cada grupo (4) el algoritmo Holotipo (4), los algoritmos LC (28) y RGC (29) para el cálculo de los conceptos, entre otros.

Después de estudiar los enfoques anteriores, en este trabajo se propone un nuevo método híbrido, no encontrado en la literatura científica revisada, siguiendo la idea básica del cálculo de los conceptos a partir del holotipo de cada grupo. La idea general del HOLOTIPO (4) se resume en que es el objeto

Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC (denominado “holotipo”) que más se parece a los restantes objeto del mismo grupo.

Siguiendo la idea general del algoritmo LC conceptual para el cálculo de los conceptos se calculan los testores típicos utilizando uno de los algoritmos existentes en la literatura consultada. Como el número de testores típicos puede ser mayor que los que se necesitan utilizar en el modelo que se propone, se calcula la utilidad de los testores típicos, como se muestra en el algoritmo 2, el cual está en función de los pesos informacionales de los rasgos presentes en éstos, como se utiliza en (6).

---

**Algoritmo 2.** Cálculo de la utilidad de los testores típicos.

**Entrada:** BC´ // BC estructurada en clases

**Salida:**  $\psi$  //utilidad de cada testor

**Paso 1:** Cálculo de los testores típicos

//se utiliza la herramienta computacional CEPAR

**Paso 2:** Calcular el peso de los rasgos que aparecen en la familia de testores típicos:

$$P(x_i) = \frac{T_i}{|T|}$$

$T_i$  : número de testores típicos donde aparece el rasgo  $i$ .

$|T|$  : número de testores típicos.

**Paso 3:** cálculo de la utilidad de los testores típicos

a. Para cada testor típico  $t_i$  calcular:

$$\psi_i(t_i) = \sum_{i=1}^{|t_i|} \varepsilon(x_i)$$

, en otras palabras, la magnitud calculada es la suma de la

Capítulo II Modelo híbrido basado en el holotipo de la clase para  
la toma de decisiones en un SBC  
importancia de los rasgos que componen  
el testor típico.

Para cada grupo se calcula el representante utilizando la idea básica del algoritmo HOLOTIPO (algoritmo 3) para a partir de éstos construir los conceptos.

**Algoritmo 3.** Cálculo del holotipo de cada grupo

**Entrada:** BC´

**Salida:** H //conjunto de holotipos ( $h_i$ ) por grupos

**Paso 1:** Determinar el típico de cada grupo:

$$t(O_i) = \frac{\beta_i}{\frac{1}{|G|-1} * \sum_{i \neq j} (\beta_i - \beta(O_i, O_j))}$$

donde  $|G|$ : cardinalidad del grupo,  $j = 1 \dots |G|$ ,  
 $\beta_i$  mayor valor de semejanza entre  $O_i$  y los  
restantes elementos del grupo y

$$\beta(O_0, O_i) = \frac{\sum_{i=1}^n \varepsilon_i \delta_i(x_i(O_i), x_i(O_0))}{\sum_{i=1}^n \varepsilon_i}$$

donde  $\varepsilon_i$ : importancia del rasgo,

$\delta_i$ : función de comparación del rasgo  $i$ .

Calculado los holotipos y la utilidad de los testores típicos se construyen los conceptos de cada clase utilizando el método híbrido propuesto en este trabajo, como se describe en el algoritmo 4.

**Algoritmo 4.** Calcular los conceptos a partir del holotipo del grupo

**Entrada:**

BC´ // Base de Casos estructurada en clases

## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

N // número de clases  
TT //conjunto de testores típicos ( $tt_i$ )  
 $\Psi$  // conjunto de utilidades ( $\psi_i$ ) asociada a cada testor típico  
H // conjunto de holotipos por clases( $h_i$ )

**Salida:** CC // conjunto de características ( $cc_i$ )

**Paso 1.** Para cada clase  $C_i$ :

- Seleccionar el  $tt_i$  de mayor  $\psi_i$  //  $i=1..p$ ,  $p$  cardinalidad de  $\Psi$
- Formar el concepto  $cc_j$  //  $j=1..r$ ,  $r$  número de testores típicos seleccionados
- Calcular la utilidad del  $cc$  obtenido utilizando la métrica  $\rho$  (Precisión del agrupamiento Grupo) propuesta en ( ):

$$\rho(c_{ij}) = \frac{|c_{ij}|}{|C_j|}$$

donde:

- ✓  $0 < \rho(c_{ij}) \leq 1$
- ✓  $|C_j|$  denota la cardinalidad del conjunto de conceptos del grupo  $j$ .  $|C_j| \neq 0$ ,  $j=1..n$ ,  $n$  cantidad de casos que conforman el grupo.
- ✓  $|c_{ij}|$  denota la cardinalidad del concepto  $i$  del grupo  $j$ .

- si  $\rho(c_{ij}) = 1$  **Entonces** el concepto  $i$  representa todos los objetos del Grupo  $j$ , es decir máxima relevancia y TERMINAR; **sino**  $PAG = PAG + \rho(c_{ij})$ , si  $i < p$  ir al paso (a.), para seleccionar el próximo  $tt$  de mayor utilidad.

---

Una vez organizada la base de casos se está en condiciones de iniciar la siguiente fase que comprende la ejecución del método de solución de problemas de los SBC. El ciclo general del RBC se basa en que, dado un nuevo caso se accede y recuperan los casos más semejantes y a partir de los mismos se adopta la decisión de la solución, ya sea la solución de uno de los casos recuperados sin modificación o aplicando una de las técnicas de adaptación existentes. En este trabajo el método de acceso, recuperación y la

---



Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC adaptación de los casos más semejantes se corresponde con la clasificación del nuevo caso, algoritmo 5.

**Algoritmo 5.** Clasificación del nuevo caso

**Entrada:**

BC' //Base de caso estructurada en clases  
 CC // Conceptos asociados a cada clase  
 O<sub>t</sub> // nuevo caso

**Salida:** C<sub>i</sub> // clase a la que O<sub>t</sub>

**Paso 1.**

a. Calcular para cada concepto de cada clase:

$$\beta(O_0, O_t) = \frac{\sum_{i=1}^n \varepsilon_i \delta_i(x_i(O_t), x_i(O_0))}{\sum_{i=1}^n \varepsilon_i} = 1$$

pertenece a dicha clase, TERMINAR

- b. Calcular la media de las semejanzas  $\beta(O_0, O_t)$  de cada concepto de cada clase y seleccionar el mayor valor.
- c. Si dicho valor es menor que 0.75 (umbral de semejanza según criterio de expertos) es un caso aislado y no se clasifica en ninguna de las clases existentes y TERMINAR.
- d. Se clasifica en la clase cuya media de las semejanzas sea mayor, TERMINAR.

---

Transcurrido por los cinco algoritmos descritos se obtiene un SBC que facilita el análisis y estudio del comportamiento de los estudiantes de la carrera de Ingeniería en Ciencias Informática en la elección de las Asignaturas Optativas que conforman el currículo optativo, lo que permite a los expertos tomar decisiones en aras de perfeccionar este importante proceso para la formación de los estudiantes.

## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

### **2.2 Metodología para el desarrollo del software HCAO**

La herramienta computacional HCAO desarrollada tiene como fundamento teórico el modelo híbrido propuesto utilizando la esencia del holotipo de una clase para el cálculo de los conceptos de éstas.

Para la implementación del software se siguen las cuatro fases de la metodología de desarrollo XP:

#### **2.2.1 Fase de Exploración**

En esta fase de la metodología XP, se realizan entrevistas con los clientes donde se planean las historias de usuario que son de su interés para la entrega del producto, las cuales posibilitan a los programadores estimar la duración del proyecto. Durante esta fase el equipo de desarrollo se familiariza con las herramientas, lenguajes y aplicaciones necesarias para el desarrollo del sistema.

#### **Lista de funcionalidades**

El proceso de captura de requisitos de la metodología de desarrollo XP se basa en identificar el listado de características que el cliente desea que existan en el sistema final. Para la herramienta computacional HCAO se determinan la siguiente lista de funcionalidades, donde algunas de las funcionalidades se ejecutan utilizando la herramienta CEPAR.

1. Cargar base de datos
2. Realizar agrupamiento
3. Calcular Matriz de Diferencia
4. Calcular Matriz Básica
5. Aplicar algoritmo de Testores Típicos
6. Calcular Matriz de Semejanza

## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

7. Mostrar objetos por grupo
8. Cargar Testores Típicos
9. Calcular importancia de los rasgos
10. Calcular utilidad de los testores típicos.
11. Calcular el holotipo de cada grupo
12. Calcular los conceptos a partir del holotipo de cada grupo
13. Clasificación del nuevo caso

### **Características del sistema**

Las características del sistema son rasgos que se desean del sistema, señalan una restricción del mismo y son fundamentales en el éxito del producto para un correcto funcionamiento. Para la herramienta se han determinado las siguientes características del sistema:

**Portabilidad:** la herramienta debe funcionar en diversas plataformas (Nova, Linux y Windows).

**Usabilidad:** la herramienta debe tener una interfaz de usuario sencilla e intuitiva, capaz de proporcionar una idea de cómo trabajar en ella, donde se debe poseer un nivel bajo o medio de conocimiento de computación.

**Hardware:** la computadora donde se usará la herramienta debe soportar procesos de complejidad algorítmica elevados: debe constar con un microprocesador P4 o superior, CPU mayor o igual a 1.5 GHz y una memoria RAM mayor o igual a 1GB.

### **Especificación de las Historias de Usuario**

Las historias de usuario (HU) son técnicas utilizadas en la metodología de desarrollo XP para representar los requisitos del software con pequeños textos en los que el cliente detalla una actividad que realizará el sistema de forma

Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC sencilla y clara, mostrando solamente el perfil de la tarea a realizarse. Para hacer más comprensible las mismas, a continuación, se describe su leyenda:


- **Número:** número de la historia de usuario incremental en el tiempo.
- **Usuario:** personas involucradas en el desarrollo de las HU.
- **Nombre de la Historia de Usuario:** nombre de la historia de usuario especificado por el programador.
- **Prioridad en negocio (Baja, Media, Alta):**
  - ✓ **Baja:** Se le otorga a las HU que son de funcionalidades auxiliares (de ayuda) y que son independientes del sistema.
  - ✓ **Media:** Se le otorga a las HU que son de funcionalidades a tener en cuenta o que resulten necesarias, pero no imprescindibles sin que estas tengan una afectación sobre el sistema que se esté desarrollando.
  - ✓ **Alta:** Se le otorga a las HU que son de funcionalidades fundamentales en el desarrollo del sistema.
- **Riesgo en desarrollo (Bajo, Medio, Alto):**
  - ✓ **Bajo:** Cuando en la implementación de las HU puedan existir errores, pero éstos son tratados fácilmente y no afectan el desarrollo del sistema.
  - ✓ **Medio:** Cuando en la implementación de las HU puedan existir errores y retrasen la entrega del producto.
  - ✓ **Alto:** Cuando en la implementación de las HU pueda existir algún error y afecte la disponibilidad del sistema.
- **Puntos estimados:** tiempo estimado que se demorará el desarrollo de la HU.

## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

- **Puntos reales:** tiempo real que se demorará el desarrollo de la HU.
- **Descripción:** breve descripción de la HU.
- **Observaciones:** Señalamiento o advertencia del sistema.
- **Interfaz gráfica:** Prototipo de interfaz solo si aplica.

Las HU son representadas mediante tablas, donde los puntos estimados y reales son expresados en semanas. A continuación, se describe una de las historias de usuario con las que cuenta el sistema (Tabla 1):

**Tabla 1. Historia de Usuario 3**

Historia de Usuario	
<b>Número:</b> HU 2	<b>Usuario:</b> Yisel Pavón Hernández Mayara Pájaro Vastillo
<b>Nombre de Historia:</b> Calcular utilidad de los testores típicos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Iteración:</b> 1	<b>Puntos Estimados:</b> 3
	<b>Puntos Reales:</b> 1
<b>Descripción:</b> la presente historia de usuario tiene como objetivo calcular la utilidad de los testores típicos.	
<b>Observaciones:</b>	
<b>Interfaz Gráfica:</b> 	

## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

### 2.2.2 Fase de Planificación

En esta fase de la metodología XP se define el alcance real del sistema. El cliente establece la prioridad de cada historia de usuario y en consecuencia los programadores realizan una estimación del esfuerzo necesario para la implementación de cada HU, además se crea el plan de iteraciones y se acuerda la posible fecha de entrega del producto.

#### – Estimación de esfuerzo de cada Historia de Usuario

Para el desempeño del proyecto es necesaria su medición, para ello se emplea la estimación por puntos o (máximo esfuerzo). Un punto es valorado como una semana de trabajo, véase tabla 2.

Tabla 2. Plan de esfuerzo por HU

Historias de Usuario	Puntos Estimados
HU 1. Calcular importancia de los rasgos	3
HU 2. Calcular utilidad de los Testores Típicos	3
HU 3. Cargar testores típicos	2
HU 4. Calcular el holotipo de cada grupo	3
HU 5. Calcular los conceptos a partir del holotipo de cada grupo	3
HU 6. Clasificación del nuevo caso	3

#### – Plan de iteraciones

Luego de ser identificadas, descritas y estimadas cada una de las HU se procede a planificar la fase de implementación del proyecto. Al finalizar cada una de las funcionalidades, éstas serán mostradas al cliente con el objetivo de detectar cambios necesarios o errores existentes. XP como parte de su ciclo de

Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC vida crea el plan de duración de iteraciones, este plan se encarga de mostrar las HU en el orden en que se implementaran en cada de una de las iteraciones, así como la duración estimada de cada una de estas. En esta etapa las HU serán desarrolladas y probadas en un ciclo iterativo, representadas en la tabla 3.

**Tabla 3. Plan de iteraciones**

Iteración	Historias de Usuario a implementar	Duración de la Iteración
1	HU 1. Calcular importancia de los rasgos HU 2. Calcular utilidad de los testores típicos HU 3. Cargar testores típicos	8 semanas
2	HU 4. Calcular Holotipo de cada grupo HU 5. Calcular los conceptos a partir del holotipo de cada grupo	6 semanas
3	HU 6. Clasificación de un nuevo caso	3 semanas

– **Plan de entregas**

En el plan de entregas se establece la fecha fin de cada iteración, en aras de lograr el producto final en la fecha establecida. Las HU que son desarrolladas para cada versión del producto y las fechas en las que se publican éstas. La tabla 4 muestra el plan de entrega para cada iteración con la planificación establecida.

**Tabla 4. Plan de entrega de las iteraciones**

Fecha final de la iteración	Historia de Usuario
Finalizada la iteración 1 31/03/2017	HU 1. Calcular importancia de los rasgos HU 2. Calcular utilidad de los testores típicos HU 3. Cargar testores típicos

## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

Finalizada la iteración 2 12/05/2017	HU 4. Calcular Holotipo de cada grupo HU 5. Calcular los conceptos a partir del holotipo de cada grupo
Finalizada la iteración 3 06/06/2017	HU 6. Clasificación de un nuevo caso

### 2.2.3 Fase de Diseño

En esta fase se realiza el diseño del sistema. La metodología ágil XP brinda una estrategia de diseño simple, realizando la implementación con el menor número de clases y métodos posibles en menor tiempo y esfuerzo.

#### – Tarjetas Clase-Responsabilidad-Colaborador

La metodología XP como parte de la fase de diseño propone el modelado de las tarjetas Clase-Responsabilidad-Colaborador (CRC), son fichas, una por cada clase donde se describe brevemente la responsabilidad de cada de clase y una lista de objetos con los que colabora para llevar a cabo esas responsabilidades, con el objetivo de desarrollar una representación organizada de clases. [19] [20] ver tabla 5.

Tabla 5. Tarjeta CRC de la clase Inicio.java

Nombre de la clase: Inicio.java	
Responsabilidad	Colaborador
Esta clase es la encargada de brindar los enlaces a la herramienta CEPAR, así como al resto de las clases existentes.	Holotipo.java, Conceptos.java

Tabla 6. Tarjeta CRC de la clase Holotipo

Nombre de la clase: Holotipo	
Responsabilidad	Colaborador
Esta clase es la encargada de mostrar el	Inicio.java, Holotipos.java



## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

cálculo del holotipo de cada grupo.	
-------------------------------------	--

**Tabla 7. Tarjeta CRC de la clase Conceptos**

<b>Nombre de la clase:</b> Holotipo	
<b>Responsabilidad</b>	<b>Colaborador</b>
Esta clase es la encargada de mostrar los conceptos de cada clase por su holotipo.	Inicio.java, Holotipo.java

### 2.3 Arquitectura del software

La arquitectura de software es la organización fundamental de un sistema en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. Es decir, la arquitectura brinda una visión global del sistema. Esto permite entenderlo, organizar su desarrollo, plantear la reutilización del software y hacerlo evolucionar (30).

Para la implementación del sistema HCAO se define un estilo arquitectónico:

- Orientado a Objetos debido a que es una aplicación de escritorio con un único rol, y no depende de la red interna o externa para su funcionamiento. Se hace necesaria la reutilización del código para su eficiencia, esto se debe al crecimiento en las posteriores fases de su desarrollo.

La Arquitectura Orientada a Objetos (AOO) facilita la incorporación de nuevos componentes y la reutilización de los existentes (31). Esta arquitectura es eficiente para muchos sistemas de software, pero está especialmente orientada para los sistemas de Desktop (25) .

Un estilo arquitectónico describe un conjunto de componentes que realiza una función requerida por el sistema. Son así mismo las unidades de modelado, diseño e implementación (32) (33).

## Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC

### 2.4 Patrones de Diseño

Los patrones de diseño se han convertido en una técnica importante para el reutilización del conocimiento de software. Cada patrón provee información sobre su diseño, describiendo las clases, métodos y relaciones que resuelven un problema de diseño en particular, estos han sido agrupados y organizados en catálogos cada uno dando diferentes clasificaciones y descripciones (34).

Un patrón de diseño es un conjunto de reglas que describen cómo afrontar tareas y solucionar problemas que surgen durante el desarrollo del software.

#### Patrones de diseño GRASP

Los patrones GRASP representan los principios básicos de la asignación de responsabilidades a objetos, expresados en forma de patrones. GRASP es el acrónimo para General Responsibility Assignment Software Patterns (Patrones Generales de Software para Asignar Responsabilidades)

- . **Experto:** se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. Dicho patrón es evidenciado en la clase Inicio.java.

- . **Alta Cohesión:** asigna una responsabilidad de forma tal que la cohesión siga siendo alta. Este patrón fue utilizado en el sistema en general al agrupar las clases en dependencia de los requisitos.

- . **Bajo Acoplamiento:** El acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. Este patrón permite realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y que a su vez permitan la

Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC reutilización. Es el mínimo de dependencias existentes entre las clases de Inicio.java y Holotipos.java, Inicio.java y Conceptos.java.

### 2.5 Patrón Arquitectónico

Los patrones arquitectónicos son los que definen la estructura de un sistema, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de tal sistema (30). Para el desarrollo del sistema se define un patrón arquitectónico:

#### - N capas

El modelo propuesto para la arquitectura de la herramienta se divide en dos capas: Vista y Clase. La capa Vista convierte y define los componentes del código en una interfaz de usuario (herramienta Desktop), que facilita al usuario interactuar con ella, y la capa Clase define los componentes del código, así como la lógica y control de su correcto funcionamiento.

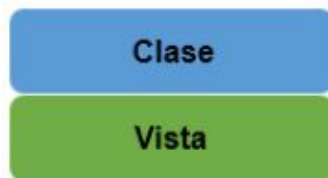


Figura 2. Esquema general de las capas del sistema

### 2.6 Consideraciones Parciales

En este capítulo se describe un modelo híbrido para la implementación de un SBC utilizando el RLCP, basado fundamentalmente en la organización de la BC estructurada en clases y representadas por el holotipo de éstas, lo que permite una organización jerárquica. Esta organización de la BC es utilizada para la implementación del método de solución de este tipo de sistema basado

Capítulo II Modelo híbrido basado en el holotipo de la clase para la toma de decisiones en un SBC en el conocimiento, lo que favorece el acceso y recuperación de los casos eficientemente.

Por otra parte, a partir de los holotipos de cada clase se calculan los conceptos, que no son más que las características distintivas de las clases lo que facilita un análisis cualitativo para la toma de decisiones. Además, se describe la arquitectura del software y los principales artefactos generados luego de aplicar la metodología de desarrollo XP a la herramienta computacional HCAO.

### **CAPÍTULO III. Pruebas e interpretación de los resultados.**

El sistema basado en casos propuesto en el capítulo 2 constituye el fundamento para el diseño y la implementación de la herramienta computacional HCAO: herramienta computacional para el análisis del comportamiento de los estudiantes en la selección de asignaturas optativas. Se exponen las características generales de este software, así como los requerimientos técnicos necesarios para su implantación.

#### **3.1 Descripción de la aplicación realizada para el análisis del comportamiento de los estudiantes en la selección de asignaturas optativas**

Para resolver un problema se necesita ante todo modelarlo y después procesar los datos. En muchos casos la solución final del problema es un programa computacional que el usuario emplea para resolver el problema en cuestión.

En este epígrafe se describe la aplicación utilizando la metodología de la modelación matemática que se propone (4) (35), la cual consta de cinco etapas:

1. Formulación del problema no matemático, es decir el problema que se quiere resolver.
2. Formalización del problema, es decir, creación del problema matemático.
3. Selección de la forma de solución del problema.
4. Solución del problema matemático.
5. Análisis e interpretación de los resultados, respecto al problema.

En la primera etapa de formulación del problema los expertos tienen una mayor participación porque son los que expresan en su lenguaje el problema a resolver. En esta etapa, se formularon los objetivos, se determinaron los casos,

### Capítulo III Prueba e interpretación de los resultados

rasgos predictivos que los describen, el rasgo objetivo, los resultados que se esperan y a partir de qué información se comienza a trabajar.

En la segunda etapa se formaliza el problema, a través de un SBC donde los casos en la BC representan los estudiantes descritos a partir de los 18 rasgos que se obtienen de la etapa de adquisición del conocimiento que es el primer paso para procesar datos.

De hecho, la forma en que se adquieran los datos es la primera transformación que se les está haciendo, es decir, la forma en que sean adquiridos los datos afectará todo el proceso posterior. Esta etapa se caracteriza por el hecho que la entrada son los datos tomados de las fuentes originales a partir de los cuales se debe extraer la información que se busca.

En este trabajo se refiere a las cuatro bases de datos con información real utilizada para la solicitud y asignación de las AO en el curso 2016-2017 a los estudiantes del tercer, cuarto y quinto año de la carrera. Así como, el intercambio directo con la comisión creada a nivel de universidad para llevar a cabo este proceso dirigida por la vicerrectoría de formación.

Asimismo se califican esos mismos miembros como expertos del sistema inteligente desarrollado. De hecho, de este trabajo de ingeniería del conocimiento entre los expertos y los ingenieros del conocimiento (autores de este trabajo) arrojó que los 8 rasgos que contiene la base de datos inicial se desglosaran en los 18 rasgos predictivos, los cuales reflejan el comportamiento o tendencia de los estudiantes al optar por las ocho AO que deben cursar en el tercer, cuarto y quinto año de la carrera.

### Capítulo III Prueba e interpretación de los resultados

La tabla 3.1 muestra los 18 rasgos predictivos que describen los casos (estudiante), el dominio y la importancia de los rasgos, a partir de criterio de expertos.

**Tabla 3.1.** Descripción de los rasgos predictores

Rasgo	Descripción	dominio	importancia
Facultad	Facultad a que pertenece el estudiante	1,2,3,4,CITEC	0.006
Índice académico	Índice académico	[2,5]	0.008
OP <sub>1</sub> , OP <sub>2</sub> , OP <sub>3</sub> , OP <sub>4</sub> , OP <sub>5</sub>	Asignatura que opta en primera opción	AD,RDN, RP, CE, ...	0.04
D <sub>1</sub> , D <sub>2</sub> , D <sub>3</sub> , D <sub>4</sub> D <sub>5</sub>	Disciplina a la que pertenece la asignatura OP <sub>i</sub>	IA, SD, IGSW P, CE, MA	0.061
IA, P, IGSW CE, MA, SD	Disciplina a la que pertenecen la asignaturas cursadas en los semestres anteriores	1,2,3,4,5,6	0.07

El rasgo objetivo se hará coincidir con la clase (grupo) al que pertenece el estudiante a partir del algoritmo de clasificación no supervisada del RLCP utilizado en el modelo propuesto.

La organización de la base de casos se define mediante una estructura jerárquica de grupos, representados por las características distintivas de los casos que los conforman calculadas a partir del elemento de mayor tipicidad de los mismos. Esta forma de organizar los casos favorece el acceso y recuperación de los casos.

Para la solución se asume que se está en presencia de un problema no supervisado, se selecciona un criterio de agrupamiento, un umbral de semejanza, se tienen en cuenta la relevancia de los rasgos predictores y se

## Capítulo III Prueba e interpretación de los resultados

aplica un algoritmo del RLCP para este tipo problema, lo cual permite agrupar los casos en clases a partir de la semejanza entre éstos.

Se calcula el holotipo de cada clase y a partir de éste se construyen los conceptos. El RBC se implementa utilizando los holotipos, dado un nuevo caso se compara con el concepto de cada clase y se selecciona la clase cuyo valor de la función de semejanza es mayor. Si este valor es igual o superior al umbral establecido se recuperan los casos de la clase seleccionada, en caso contrario se considera un caso aislado y el sistema no puede tomar ninguna decisión.

A partir, de los casos recuperados se utiliza para ofrecer la solución el de mayor grado de semejanza con el nuevo caso y los restantes sirven de apoyo al módulo de explicación para argumentar la solución dada.

### **3.2 Análisis e interpretación de los resultados**

Como resultado de aplicar un algoritmo de agrupamiento no supervisado, tomando como criterio un Bo- Compacto, un umbral de semejanza  $\beta_0=0.52$  se formaron 43 clases de un total de 652 casos.

Una clase agrupa 582 casos, otros 17 casos y los restantes casos están distribuidos por las 42 clases restantes. Lo que es interpretado por los expertos que la información almacenada conlleva a que existe una tendencia de un comportamiento similar de los estudiantes para optar por las asignaturas del currículo optativo o es necesario tener en cuenta más información principalmente la relacionada con los rasgos predictores que permita otros criterios para diferenciar los estudiantes.

No obstante, cuando el umbral de semejanza  $\beta_0$  se calcula utilizando otra expresión o por criterio de experto cuyo valor es inferior el número de clases



## Capítulo III Prueba e interpretación de los resultados

disminuye y la distribución de objetos por éstas tiene menor grado de dispersión.

Para mejor comprensión de la importancia de los conceptos en el análisis cualitativo para la toma de decisiones se explica el concepto de la clase 1 que resultó ser la de mayor cardinalidad. El caso que representa esta clase, holotipo, que es el objeto que más se parece a todos en el grupo, se puede interpretar como: estudiantes de la Facultad 4, de cuarto año académico, con un índice académico de 4.4 y con tendencia a solicitar en las primeras dos opciones asignaturas de la Disciplina Ciencias Empresariales, con énfasis en las asignaturas Comercio electrónico Organizacional.

La clase 2 que es la de segunda mayor cardinalidad representa estudiantes de la Facultad 1, del quinto año de la carrera, con un índice académico de 4.5 con tendencia a solicitar en las dos primeras opciones asignaturas de la Disciplina Técnicas de Programación y Sistemas Digitales, particularmente en las asignaturas Análisis y Diseño de Algoritmos e Informática Forense y Hacking Ético respectivamente.

Para un valor de umbral de semejanza  $\beta_0=0.33$  calculado por criterio de expertos se obtienen 14 clases con una distribución más uniforme de los casos, cuya mediana (Sampieri 2006) de la cardinalidad de los grupos de 32 objetos.

### **3.3 Características generales de la herramienta HCAO**

Las figuras 3.1, 3.2 y 3.3 ilustran imágenes de la herramienta HCAO implementadas.



Figura 3.1. Interfaz Principal

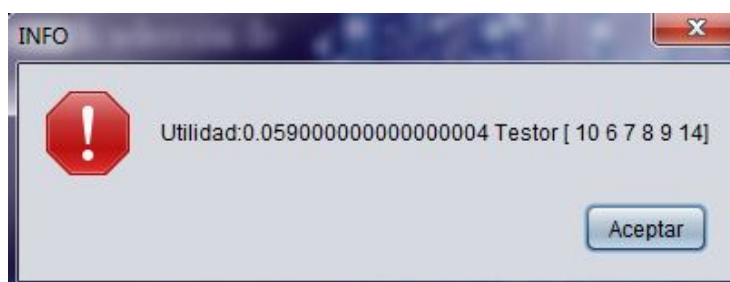


Figura 3.2. Utilidad del Testor Típico

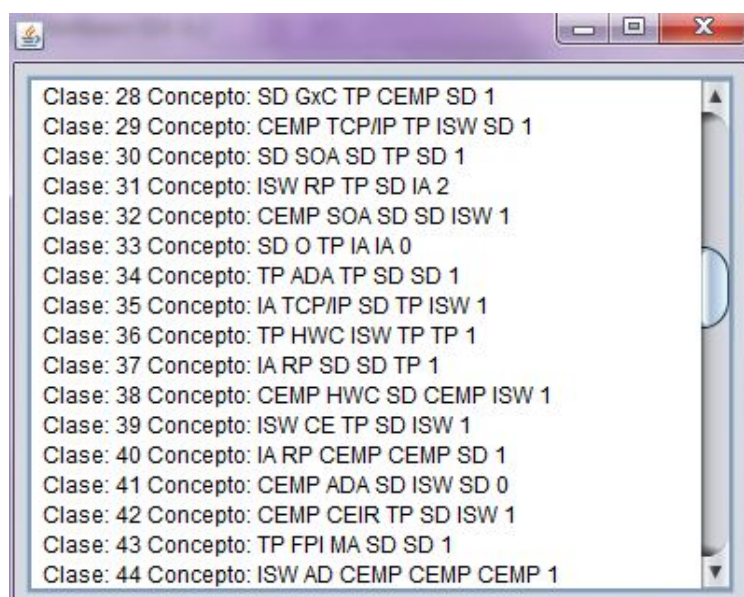


Figura 3.3. Cálculo del Holotipo

## Capítulo III Prueba e interpretación de los resultados

### 3.4 Fase de Prueba

En este capítulo se muestran las pruebas realizadas al sistema para comprobar su correcto funcionamiento. Dentro de las fases propuestas por la metodología de desarrollo XP se lleva a cabo las pruebas del sistema. Esta metodología divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación.

#### – Estrategia de Prueba: (Pruebas Unitarias)

Nivel de Prueba: pruebas de desarrollador

Tipo de Prueba: funcionalidad

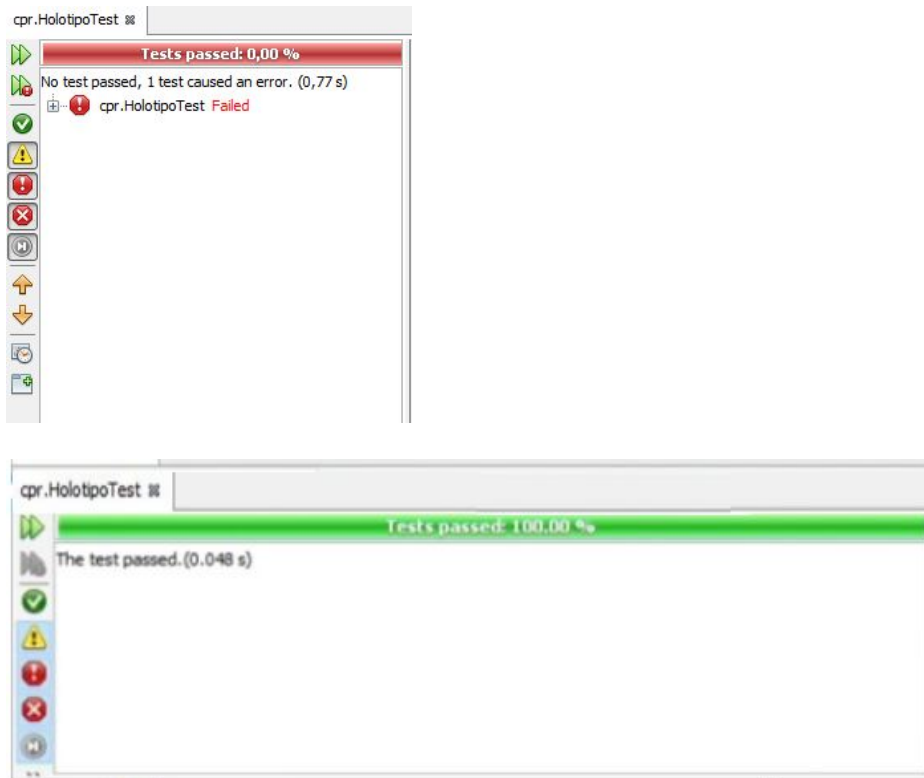
Método de Prueba: pruebas de caja blanca

Herramienta Utilizada: JUnit (Método que prueba una unidad de código)

Para automatizar las pruebas unitarias se utilizó una librería JUnit, que permite realizar la ejecución de clases Java de manera controlada, esta evalúa si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera.

Las imágenes que a continuación se muestran es el resultado de las pruebas realizadas a la clase Holotipo.

## Capítulo III Prueba e interpretación de los resultados



Una vez detectados los errores se realizaron las validaciones. La Figura 3.4 muestra un resumen del comportamiento de los errores encontrados los cuales fueron corregidos en tres iteraciones.

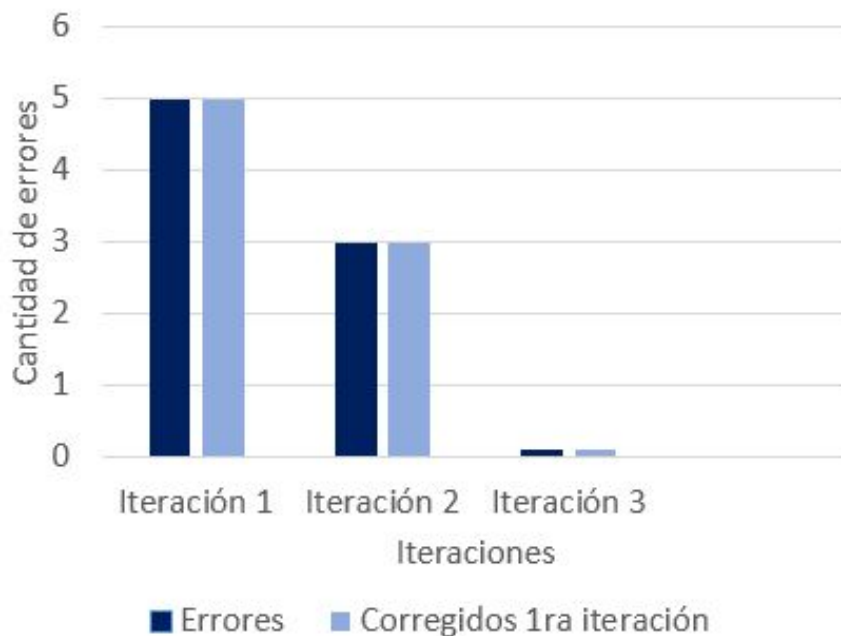


Figura 3.4. Resultado de las pruebas unitarias.

- Estrategias de pruebas: (Pruebas de Aceptación)

## Capítulo III Prueba e interpretación de los resultados

Nivel de prueba: pruebas del sistema

Tipo de Prueba: funcionalidad

Método de prueba: pruebas de caja negra

Técnica o herramienta utilizada: partición de equivalencia (conjunto de juegos de datos que tiene tres comportamientos válidos, inválidos e incompletos).

### Matriz de equivalencia.

Tabla 3.2. Matriz de equivalencia

Identificador	Entrada	Clases validadas	Clases invalidas	Clases incompletas
Clasificación de un nuevo caso	rasgos	Todos los campos con los rasgos correctos	Algún campo con un rasgo incorrecto	Al menos un campo vacío

### 3.5 Consideraciones Parciales

En este capítulo con la experimentación realizada se demuestra que se puede aplicar un algoritmo de Reconocimiento Lógico Combinatorio de Patrones sobre la base de datos, para obtener información relevante que sirva de apoyo a la toma de decisiones.

Con la realización de las pruebas unitarias se comprobó el correcto funcionamiento de las características funcionales de la herramienta, logrando los resultados esperados.

Con la aplicación de las pruebas de aceptación se obtuvo la satisfacción del usuario, comprobando que las funcionalidades responden a las acciones que realiza en la herramienta.

### **CONCLUSIONES GENERALES**

El estudio del estado arte realizado permite determinar las factibilidades de aplicar técnicas del RLCP para extraer la información relevante del comportamiento de los estudiantes en la selección de AO de las bases de datos de los últimos tres años académicos de la carrera de Ingeniería en Ciencias Informáticas.

Además, la utilización del algoritmo de agrupamiento no supervisado Holotipo permite la organización jerárquica de la base de casos lo que favorece el acceso y recuperación de los casos más semejantes de manera más eficiente.

Por otra parte, el desarrollo de la herramienta computacional HCAO facilita a los expertos la comprensión de la información existente en la base de datos contribuyendo así, a la toma de decisiones para el perfeccionamiento del currículo optativo del Plan de Estudios de la carrera.

Los resultados de las pruebas de software realizadas permiten comprobar el correcto funcionamiento de la herramienta desarrollada.

### **RECOMENDACIONES**

A partir de la investigación realizada y de las conclusiones generales emanadas, se recomienda:

- Incorporar funcionalidades que permitan establecer comparaciones entre los resultados arrojados con el algoritmo propuesto y otros del estado del arte.
- Incorporar funcionalidades que faciliten la visualización de los resultados para una mejor comprensión.

**REFERENCIAS**

1. **Bello, R.** *Aplicaciones de la Inteligencia Artificial*. Jalisco : Ediciones de La noche, 2002.
2. **Rich, E.** *Inteligencia Artificial*. Barcelona : Gustavo Gili S.A, 1988.
3. **Martínez Sánchez, N, García Valdivia, Z Z y Garcia Lorenzo, MM.** Modelo para diseñar Sistemas de Enseñanza-Aprendizaje Inteligentes utilizando el Razonamiento Basado en Casos. *Tesis presentada en opción del grado científico de Doctor en Ciencias Técnicas*. s.l., Villa Clara, Cuba : UCLV, 2009.
4. **Ruiz-Shulcloper, J.** *Reconocimiento Lógico Combinatorio de Patrones: Teoría de aplicaciones*. Centro de Aplicaciones de Tecnologías de Avanzada. 2009. Tesis en opción al grado científico de doctor en Ciencias.
5. **Pons Porrata, A.** *Desarrollo de algoritmo para la estructuración dinámica de información y su aplicación a la detección de sucesos*. 2004. Trabajo de tesis en opción al grado científico de Doctor en Ciencias Técnicas.
6. **González, Yunia, Martinez, Natalia y Díaz, Adolfo.** *Modelo para la Adaptación de las soluciones en un sistema basado en casos utilizando el agrupamiento conceptual*. La Habana : s.n., 2014.
7. *El agrupamiento conceptual en el concepto de la teoría de los conjuntos aproximados.* **Gonzalez, Yunia y Martínez, Natalia.** 1, s.l.: Dyna New Technologies, 2015, Vol. 2, págs. 1-12.
8. **González, Yunia.** Métricas para la validación de los conceptos en el reconocimiento lógico combinatorio de patrones. *III Conferencia Internacional en Ciencias Computacionales*. CICCI, 2016.
9. **Sampieri, R H, Collado, C F y Lucio, P B.** *Metodología de la Investigación*. McGraw-Hill Interamericana de México : s.n., 2006.
10. **Kolodner, J L.** *An Introduction to case-base Reasoning*. *Artificial Intelligence Review* 6. 1992. págs. 3-34.
11. **López de Mántaras, R.** *Retrieval, reuse, revision, and retention in case based reasoning*. [ed.] *The Knowledge Engineering Review*. 2005. Vols. 00: 1-2.
12. **Rodríguez, Y y García, M.** *Generalización de la métrica basada en la diferencia de valores (VDM) para variables lingüísticas y su aplicación en sistemas basados en el conocimiento*. 2007. Tesis presentada en opción del grado científico de Doctor en Ciencias Técnicas.
13. *LEX: Un nuevo algoritmo para el cálculo de los testores Típicos.* **Pons, A.** Cuba : s.n., 2003, Ciencias Matemáticas.
14. *Two new feacture selection algorithms with Rough Sets Theory.* **Caballero, Y y Bello, R.** 217:209-216, Boston : s.n., 2006, Vols. *Artificial Intelligence in theory and Practice*. M. Bramer, Springer.
15. *An overview of the evolution of the concept of testor.* **Lazo, M, Ruiz, J y Otros.** 753-762, 2001, Vol. *Pattern Recognition*(34).
16. **lias, Rodriguez A y Sanchez , diaz G.** *An Algorithm for Computing Typical Testors Based on Elimination of Gaps and Reduction of columns (FAST BR)*. 2013. Vol. 27.
17. **Lias Rodriguez, Alexsey y Pons Porrata, Aurora.** *Un nuevo Algoritmo de escalar para el cálculo de los testores típicos(BR)*.
18. *Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts*.



- Journal of Policy Analysis and Information Systems*. **Michalski, R S.** 3, 1980, Vol. 4. 219-244.
19. **Laza, R, Fernandez, F y Corchado, J M.** *Sistema Basado en Casos para el Soporte a la Toma de Decisiones*. Informática de Gestión, Universidad de Vigo. España : Campus As Lagoas.
20. **Grass Boada, Darian Horacio, y otros.** *TÉCNICAS DE AGRUPAMIENTO APLICADAS A LOS REGISTROS DE NAVEGACIÓN POR INTERNET*.
21. **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.
22. **Corso, C L.** *Aplicación de Algoritmo de clasificación supervisada usando WEKA*.
23. **Yero-Oses, E A, Reyes González, Yunia y Martínez Sánchez, N.** *CEPAR:Un sistema herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones*. 2016. Peña Tecnológica.
24. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico*. quinta. Madrid : Concepcion Fernández, 2002.
25. —. *Software Engineering, a practitioner's approach*. . s.l. : 7th edición, 2011.
26. **Gonzalez Miranda, Elismailen Alicia y Gonzalez Fernandez, Jorge Adrian.** *Herramienta computacional para los Sistemas Integrados de Gestión Bibliotecaria utilizando algoritmos conceptuales del Reconocimiento Lógico combinatorio de Patrones*. La Habana : s.n., 2016. Tesis.
27. **Netbeans.** [En línea] [http://netbeans.org/..](http://netbeans.org/)
28. **Trinidad, Fransisco M.** *Herramienta para la estructuración conceptual de espacios*. 2001. Vol. 4.
29. **Pons Porrata, A.** *RGC: A new conceptual clustering algorithm for mixed incomplete data sets*. 2002.
30. **Almeira, Adriana Sandra y Pérez Cavenago, Vanina.** *Tesis de Arquitectura de Software: Estilos y Patrones*. Argentina : s.n., 2007.
31. **Durillo, J J.** *Arquitectura Orientada a Objetos*. 2010.
32. [En línea] <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
33. **ECURED.** [En línea] [https://www.ecured.cu/Estilos\\_arquitect%C3%B3nicos](https://www.ecured.cu/Estilos_arquitect%C3%B3nicos).
34. **Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel.** *Arquitecturas de Software*. 2004.
35. **Ruiz Shulcloper, J, Carrasco Ochoa, J A y Martinez Trinidad, J F.** *Reconocimiento de patrones:conceptos y metodología*. s.l. : CENATAV, 2013.
36. **Felinfo.** *Java, Linux, Virtualización. Open Source*. [En línea]. [En línea] 2010. <http://felinfo.blogspot.com/2010/05/leer-un-fichero-excel-desde-java.html..>

**BIBLIOGRAFÍA**

1. Bello, R. *Aplicaciones de la Inteligencia Artificial*. Jalisco : Ediciones de La noche, 2002.
2. Rich, E. *Inteligencia Artificial*. Barcelona : Gustavo Gili S.A, 1988.
3. Martínez Sánchez, N, García Valdivia, Z Z y Garcia Lorenzo, MM. *Modelo para diseñar Sistemas de Enseñanza-Aprendizaje Inteligentes utilizando el Razonamiento Basado en Casos*. Tesis presentada en opción del grado científico de Doctor en Ciencias Técnicas. s.l., Villa Clara, Cuba : UCLV, 2009.
4. Ruiz-Shulcloper, J. *Reconocimiento Lógico Combinatorio de Patrones: Teoría de aplicaciones*. Centro de Aplicaciones de Tecnologías de Avanzada. 2009. Tesis en opción al grado científico de doctor en Ciencias.
5. Pons Porrata, A. *Desarrollo de algoritmo para la estructuración dinámica de información y su aplicación a la detección de sucesos*. 2004. Trabajo de tesis en opción al grado científico de Doctor en Ciencias Técnicas.
6. González, Yunia, Martínez, Natalia y Díaz, Adolfo. *Modelo para la Adaptación de las soluciones en un sistema basado en casos utilizando el agrupamiento conceptual*. La Habana : s.n., 2014.
7. *El agrupamiento conceptual en el concepto de la teoría de los conjuntos aproximados*. Gonzalez, Yunia y Martínez, Natalia. 1, s.l. : Dyna New Technologies, 2015, Vol. 2, págs. 1-12.
8. González, Yunia. *Métricas para la validación de los conceptos en el reconocimiento lógico combinatorio de patrones*. III Conferencia Internacional en Ciencias Computacionales. CICCI, 2016.
9. Sampieri, R H, Collado, C F y Lucio, P B. *Metodología de la Investigación*. McGraw-Hill Interamericana de México : s.n., 2006.
10. Kolodner, J L. *An Introduction to case-base Reasoning*. *Artificial Intelligence Review* 6. 1992. págs. 3-34.
11. López de Mántaras, R. *Retrieval, reuse, revision, and retention in case based reasoning*. [ed.] *The Knowledge Engineering Review*. 2005. Vols. 00: 1-2.
12. Rodríguez, Y y García, M. *Generalización de la métrica basada en la diferencia de valores (VDM) para variables lingüísticas y su aplicación es sistemas basados en el conocimiento*. 2007. Tesis presentada en opción del grado científico de Doctor en Ciencias Técnicas.
13. *LEX: Un nuevo algoritmo para el cálculo de los testores Típicos*. Pons, A. Cuba : s.n., 2003, Ciencias Matemáticas.
14. *Two new feacture selection algorithms with Rough Sets Theory*. Caballero, Y y Bello, R. 217:209-216, Boston : s.n., 2006, Vols. *Artificial Intelligence in theory and Practice*. M. Bramer, Springer.
15. *An overview of the evolution of the concept of testor*. Lazo, M, Ruiz, J y Otros. 753-762, 2001, Vol. *Pattern Recognition*(34).
16. *lias, Rodriguez A y Sanchez , diaz G. An Algorithm for Computing Typical Testors Based on Elimination of Gaps and Reduction of columns (FAST BR)*. 2013. Vol. 27.
17. *Lias Rodriguez, Alexsey y Pons Porrata, Aurora. Un nuevo Algoritmo de escalar para el cálculo de los testores típicos(BR)*.
18. *Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive*

- concepts. *Journal of Policy Analysis and Information Systems*. Michalski, R S. 3, 1980, Vol. 4. 219-244.
19. Laza, R, Fernandez, F y Corchado, J M. *Sistema Basado en Casos para el Soporte a la Toma de Decisiones*. *Informática de Gestión, Universidad de Vigo*. España : Campus As Lagoas.
20. Grass Boada, Darian Horacio, y otros. **TÉCNICAS DE AGRUPAMIENTO APLICADAS A LOS REGISTROS DE NAVEGACIÓN POR INTERNET**.
21. 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.
22. Corso, C L. *Aplicación de Algoritmo de clasificación supervisada usando WEKA*.
23. Yero-Oses, E A, Reyes González, Yunia y Martínez Sánchez, N. *CEPAR:Un sistema herramienta de apoyo al docente del reconocimiento lógico combinatorio de patrones*. 2016. Peña Tecnológica.
24. Pressman, Roger S. *Ingeniería de Software. Un enfoque práctico*. quinta. Madrid : Concepcion Fernández, 2002.
25. —. *Software Engineering, a practitioner's approach*. . s.l. : 7th edición, 2011.
26. Gonzalez Miranda, Elismailen Alicia y Gonzalez Fernandez, Jorge Adrian. *Herramienta computacional para los Sistemas Integrados de Gestión Bibliotecaria utilizando algoritmos conceptuales del Reconocimiento Lógico combinatorio de Patrones*. La Habana : s.n., 2016. Tesis.
27. Netbeans. [En línea] <http://netbeans.org/>.
28. Trinidad, Fransisco M. *Herramienta para la estructuración conceptual de espacios*. 2001. Vol. 4.
29. Pons Porrata, A. *RGC: A new conceptual clustering algorithm for mixed incomplete data sets*. 2002.
30. Almeida, Adriana Sandra y Pérez Cavenago, Vanina. *Tesis de Arquitectura de Software: Estilos y Patrones*. Argentina : s.n., 2007.
31. Durillo, J J. *Arquitectura Orientada a Objetos*. 2010.
32. [En línea] <http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>.
33. ECURED. [En línea] [https://www.ecured.cu/Estilos\\_arquitect%C3%B3nicos](https://www.ecured.cu/Estilos_arquitect%C3%B3nicos).
34. Camacho, Erika, Cardeso, Fabio y Nuñez, Gabriel. *Arquitecturas de Software*. 2004.
35. Ruiz Shulcloper, J, Carrasco Ochoa, J A y Martinez Trinidad, J F. *Reconocimiento de patrones:conceptos y metodología*. s.l. : CENATAV, 2013.
36. Felinfo. *Java, Linux, Virtualización. Open Source*. [En línea]. [En línea] 2010. <http://felinfo.blogspot.com/2010/05/leer-un-fichero-excel-desde-java.html>.
37. <http://redyseguridad.fip.unam.mx/proyectos/biometria/basesteoricas/reconocimiento.html>
38. [https://ccrma.stanford.edu/~juanig/articles/pidht/pidtoot/Reconocimiento\\_Patrones.html](https://ccrma.stanford.edu/~juanig/articles/pidht/pidtoot/Reconocimiento_Patrones.html)

39. **Enfoque lógico combinatorio al reconocimiento de patrones I. Selección de variables y clasificación supervisada\_files**  
<https://www.iberlibro.com/Enfoque-l%C3%B3gico-combinatorio-reconocimiento-patrones-Selecci%C3%B3n/11387224527/bd>
40. <http://www.ejournal.unam.mx/cys/vol04-01/CYS04107.pdf>
41.  
<https://www.infor.uva.es/~calonso/IAI/TrabajoAlumnos/Razonamiento%20basado%20en%20casos.pdf>
42. **ECURED.** [En línea]  
[https://www.ecured.cu/Sistemas\\_basados\\_en\\_reglas](https://www.ecured.cu/Sistemas_basados_en_reglas).
43. (Comparativa metodologías ágiles vs tradicionales)  
<http://www.avante.es/comparativa-metodologias-agiles-vs-tradicionales/>
44. (Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)) <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
45. (METODOLOGIAS DE DESARROLLO DE SOFTWARE TRADICIONALES VS AGILES)  
<http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>
46. (Metodologías de Desarrollo Ágiles Vs. Metodologías Tradicionales)  
<http://rdsoporteymantenimientodepc.blogspot.com/2014/03/metodologias-de-desarrollo-agiles-vs.html>
47. (METODOLOGÍAS TRADICIONALES)  
<https://www.clubensayos.com/Temas-Variados/METODOLOG%C3%8DAS-TRADICIONALES/412898.html>
48. <https://prezi.com/xpo8pdv2ghqm/metodologias-tradicionales-vs-metodologias-agiles/>
49. [http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html)
50. LARMAN, C. UML y Patrones. New York : Prentice Hall, 1999
51. jaba 46 Jendrock, E. et al. Java Platform, Enyerprise Edition. 2014
52. netbeans 47 Mendoza-González, G. Herramienta de Desarrollo Netbeans. 2015.
53. [estilos.pdf](http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF)  
<http://carlosreynoso.com.ar/archivos/arquitectura/Estilos.PDF>
54. Introducción al Patrón de Arquitectura por Capas – María Eugenia Arevalo Lizardo  
<https://arevalomaria.wordpress.com/2010/12/02/introduccion-al-patron-de-arquitectura-por-capas>

## Anexos

## Anexo1. Historia de Usuario

Tabla 9. Historia de Usuario 1

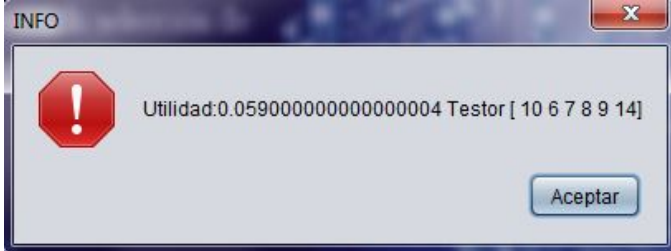
Historia de Usuario	
<b>Número:</b> HU 1	<b>Usuario:</b> Yisel Pavón Hernández Mayara Pájaro Vastillo
<b>Nombre de Historia:</b> Cargar Testores Típicos	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Medio
<b>Iteración:</b> 1	<b>Puntos Estimados:</b> 2
	<b>Puntos Reales:</b> 1
<b>Descripción:</b> la presente historia de usuario tiene como objetivo cargar y mostrar el cálculo de los Testores Típicos.	
<b>Observaciones:</b>	
<b>Interfaz Gráfica:</b>	
	

Tabla 10. Historia de Usuario 2

Historia de Usuario	
<b>Número:</b> HU 2	<b>Usuario:</b> Yisel Pavón Hernández Mayara Pájaro Vastillo
<b>Nombre de Historia:</b> Calcular importancia de los rasgos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alto
<b>Iteración:</b> 1	<b>Puntos Estimados:</b> 3
	<b>Puntos Reales:</b> 1
<b>Descripción:</b> la presente historia de usuario tiene como objetivo calcular la importancia de los rasgos	
<b>Observaciones:</b>	

Interfaz Gráfica:

Tabla 11. Historia de Usuario 4

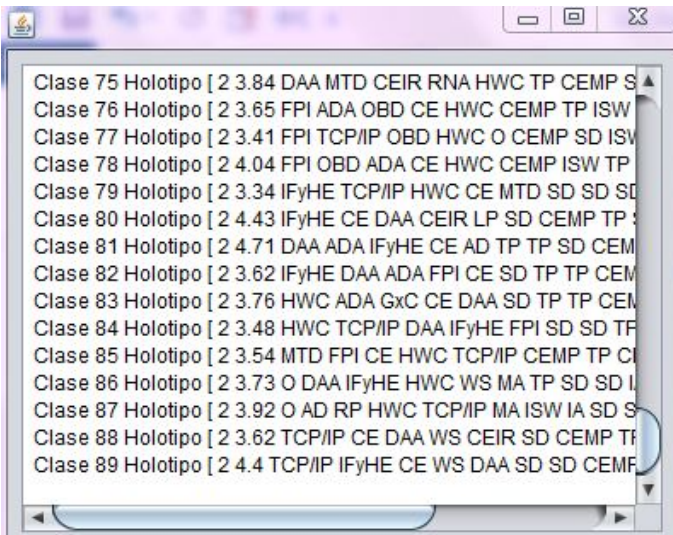
Historia de Usuario	
<b>Número:</b> HU 4	<b>Usuario:</b> Yisel Pavón Hernández Mayara Pájaro Vastillo
<b>Nombre de Historia:</b> Calcular el Holotipo de cada grupo	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alto
<b>Iteración:</b> 3	<b>Puntos Estimados:</b> 3
	<b>Puntos Reales:</b> 2
<b>Descripción:</b> la presente historia de usuario tiene como objetivo calcular y mostrar el holotipo de cada grupo.	
<b>Observaciones:</b>	
<b>Interfaz Gráfica:</b>  <p>The screenshot shows a window with a list of classes. Each entry includes a class number, the word 'Holotipo', a numerical value, and a series of alphanumeric tags. The list is as follows:</p> <ul style="list-style-type: none"> <li>Clase 75 Holotipo [ 2 3.84 DAA MTD CEIR RNA HWC TP CEMP S</li> <li>Clase 76 Holotipo [ 2 3.65 FPI ADA OBD CE HWC CEMP TP ISW</li> <li>Clase 77 Holotipo [ 2 3.41 FPI TCP/IP OBD HWC O CEMP SD ISW</li> <li>Clase 78 Holotipo [ 2 4.04 FPI OBD ADA CE HWC CEMP ISW TP</li> <li>Clase 79 Holotipo [ 2 3.34 IFyHE TCP/IP HWC CE MTD SD SD SD</li> <li>Clase 80 Holotipo [ 2 4.43 IFyHE CE DAA CEIR LP SD CEMP TP S</li> <li>Clase 81 Holotipo [ 2 4.71 DAA ADA IFyHE CE AD TP TP SD CEM</li> <li>Clase 82 Holotipo [ 2 3.62 IFyHE DAA ADA FPI CE SD TP TP CEM</li> <li>Clase 83 Holotipo [ 2 3.76 HWC ADA GxC CE DAA SD TP TP CEM</li> <li>Clase 84 Holotipo [ 2 3.48 HWC TCP/IP DAA IFyHE FPI SD SD TP</li> <li>Clase 85 Holotipo [ 2 3.54 MTD FPI CE HWC TCP/IP CEMP TP C</li> <li>Clase 86 Holotipo [ 2 3.73 O DAA IFyHE HWC WS MA TP SD SD I</li> <li>Clase 87 Holotipo [ 2 3.92 O AD RP HWC TCP/IP MA ISW IA SD S</li> <li>Clase 88 Holotipo [ 2 3.62 TCP/IP CE DAA WS CEIR SD CEMP TP</li> <li>Clase 89 Holotipo [ 2 4.4 TCP/IP IFyHE CE WS DAA SD SD CEMP</li> </ul>	

Tabla 12. Historia de Usuario 5

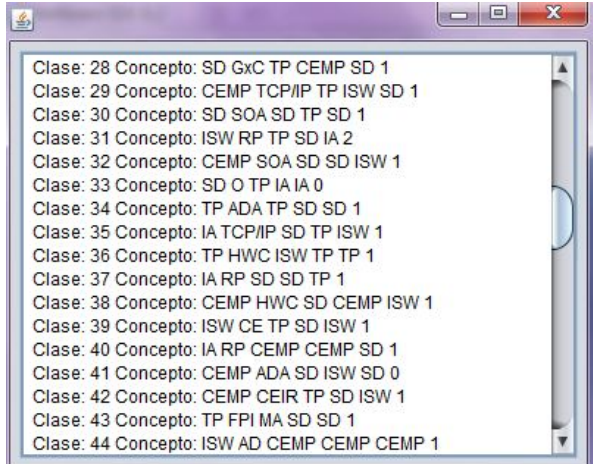
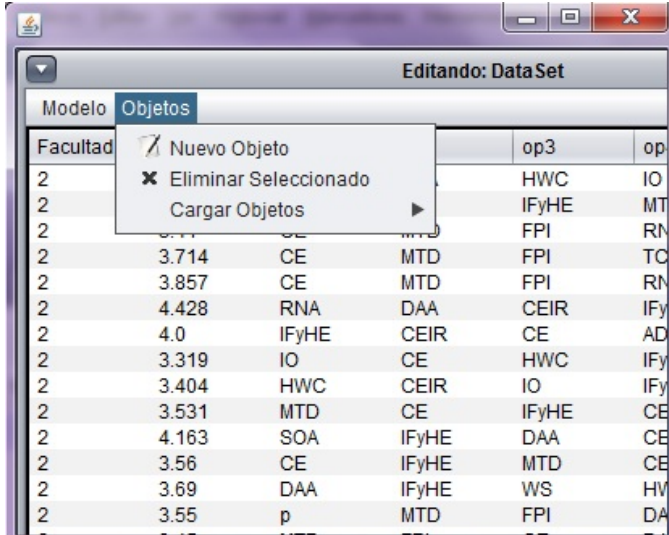
Historia de Usuario	
<b>Número:</b> HU 5	<b>Usuario:</b> Yisel Pavón Hernández Mayara Pájaro Vastillo
<b>Nombre de Historia:</b> Calcular los conceptos a partir del holotipo de cada grupo	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alto
<b>Iteración:</b> 3	<b>Puntos Estimados:</b> 3
	<b>Puntos Reales:</b> 2
<b>Descripción:</b> la presente historia de usuario tiene como objetivo calcular y mostrar los conceptos a partir del cálculo del Holotipo de cada grupo.	
<b>Observaciones:</b>	
<b>Interfaz Gráfica:</b>	
 <p>The screenshot shows a window with a list of classes and concepts. The text in the window is as follows:</p> <pre> Clase: 28 Concepto: SD GxC TP CEMP SD 1 Clase: 29 Concepto: CEMP TCP/IP TP ISW SD 1 Clase: 30 Concepto: SD SOA SD TP SD 1 Clase: 31 Concepto: ISW RP TP SD IA 2 Clase: 32 Concepto: CEMP SOA SD SD ISW 1 Clase: 33 Concepto: SD O TP IA IA 0 Clase: 34 Concepto: TP ADA TP SD SD 1 Clase: 35 Concepto: IA TCP/IP SD TP ISW 1 Clase: 36 Concepto: TP HWC ISW TP TP 1 Clase: 37 Concepto: IA RP SD SD TP 1 Clase: 38 Concepto: CEMP HWC SD CEMP ISW 1 Clase: 39 Concepto: ISW CE TP SD ISW 1 Clase: 40 Concepto: IA RP CEMP CEMP SD 1 Clase: 41 Concepto: CEMP ADA SD ISW SD 0 Clase: 42 Concepto: CEMP CEIR TP SD ISW 1 Clase: 43 Concepto: TP FPI MA SD SD 1 Clase: 44 Concepto: ISW AD CEMP CEMP CEMP 1 </pre>	

Tabla 13. Historia de Usuario 6

Historia de Usuario																																																													
<b>Número:</b> HU 6	<b>Usuario:</b> Yisel Pavón Hernández Mayara Pájaro Vastillo																																																												
<b>Nombre de Historia:</b> Clasificación del nuevo caso																																																													
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Medio																																																												
<b>Iteración:</b> 4	<b>Puntos Estimados:</b> 3																																																												
	<b>Puntos Reales:</b> 2																																																												
<b>Descripción:</b> la presente historia de usuario tiene como objetivo clasificar un nuevo caso.																																																													
<b>Observaciones:</b>																																																													
<b>Interfaz Gráfica:</b>																																																													
 <p>The screenshot shows a window titled 'Editando: Data Set'. It contains a table with columns 'Modelo', 'Objetos', 'op3', and 'op'. A context menu is open over the table, showing options: 'Nuevo Objeto', 'Eliminar Seleccionado', and 'Cargar Objetos'. The table data is as follows:</p> <table border="1"> <thead> <tr> <th>Modelo</th> <th>Objetos</th> <th>op3</th> <th>op</th> </tr> </thead> <tbody> <tr><td>2</td><td></td><td>HWC</td><td>IO</td></tr> <tr><td>2</td><td></td><td>IFyHE</td><td>MT</td></tr> <tr><td>2</td><td></td><td>FPI</td><td>RN</td></tr> <tr><td>2</td><td>3.714 CE MTD</td><td>FPI</td><td>TC</td></tr> <tr><td>2</td><td>3.857 CE MTD</td><td>FPI</td><td>RN</td></tr> <tr><td>2</td><td>4.428 RNA DAA</td><td>CEIR</td><td>IFy</td></tr> <tr><td>2</td><td>4.0 IFyHE CEIR</td><td>CE</td><td>AD</td></tr> <tr><td>2</td><td>3.319 IO CE</td><td>HWC</td><td>IFy</td></tr> <tr><td>2</td><td>3.404 HWC CEIR</td><td>IO</td><td>IFy</td></tr> <tr><td>2</td><td>3.531 MTD CE</td><td>IFyHE</td><td>CE</td></tr> <tr><td>2</td><td>4.163 SOA IFyHE</td><td>DAA</td><td>CE</td></tr> <tr><td>2</td><td>3.56 CE IFyHE</td><td>MTD</td><td>CE</td></tr> <tr><td>2</td><td>3.69 DAA IFyHE</td><td>WS</td><td>HV</td></tr> <tr><td>2</td><td>3.55 p MTD</td><td>FPI</td><td>DA</td></tr> </tbody> </table>		Modelo	Objetos	op3	op	2		HWC	IO	2		IFyHE	MT	2		FPI	RN	2	3.714 CE MTD	FPI	TC	2	3.857 CE MTD	FPI	RN	2	4.428 RNA DAA	CEIR	IFy	2	4.0 IFyHE CEIR	CE	AD	2	3.319 IO CE	HWC	IFy	2	3.404 HWC CEIR	IO	IFy	2	3.531 MTD CE	IFyHE	CE	2	4.163 SOA IFyHE	DAA	CE	2	3.56 CE IFyHE	MTD	CE	2	3.69 DAA IFyHE	WS	HV	2	3.55 p MTD	FPI	DA
Modelo	Objetos	op3	op																																																										
2		HWC	IO																																																										
2		IFyHE	MT																																																										
2		FPI	RN																																																										
2	3.714 CE MTD	FPI	TC																																																										
2	3.857 CE MTD	FPI	RN																																																										
2	4.428 RNA DAA	CEIR	IFy																																																										
2	4.0 IFyHE CEIR	CE	AD																																																										
2	3.319 IO CE	HWC	IFy																																																										
2	3.404 HWC CEIR	IO	IFy																																																										
2	3.531 MTD CE	IFyHE	CE																																																										
2	4.163 SOA IFyHE	DAA	CE																																																										
2	3.56 CE IFyHE	MTD	CE																																																										
2	3.69 DAA IFyHE	WS	HV																																																										
2	3.55 p MTD	FPI	DA																																																										