

Universidad de Sancti Spíritus  
"José Martí Pérez"  
Facultad de Informática  
Carrera de Ingeniería Informática



## Trabajo de diploma para optar por el título de Ingeniería Informática

**Tema:** "Sistema Experto para contribuir al Proceso de Enseñanza-Aprendizaje de la Geometría Plana y del Espacio de los estudiantes de 10mo grado en el IPU Nieves Morejón López de Cabaiguán"

**Autor(a):** Geidy Cabrera Pérez.

**Tutor(a):** Dra.C Lydia Rosa Ríos Rodríguez.

**Consultante:** MsC. Zulema Pérez Pérez.

**Junio de 2014**

**Curso 2013-2014**

## Agradecimientos

*A mi madre por todo el amor, por su constante preocupación y las fuerzas que siempre me ha dado cuando más lo necesito. Te quiero*

*A mi padre querido que siempre estuvo a mi lado en cada paso de mi vida y me apoyó incondicionalmente. Te adoro*

*A mi hermana que siempre me ayudó.*

*A mis abuelos, tíos y tías por su apoyo diario.*

*A todos mis compañeros de grupo por su ayuda durante estos cinco años de experiencias compartidas.*

*A mis amigos Carlos y Ariamna, gracias por aceptarme como soy.*

*A mi tutora que con su experiencia, tiempo y dedicación me ha ayudado a cumplir mi sueño, graduarme*

*A todos mis profesores porque de una manera u otra me han brindado sus conocimientos y experiencias.*

*En general a todos los que participaron en la investigación de mi trabajo, a los que me animaron a seguir adelante y a todos aquellos que han hecho posible este momento.*

---

## **Dedicatoria**

*A todos mis seres queridos por apoyarme durante cada etapa en mi vida, por brindarme la seguridad para seguir adelante y porque sin ustedes no hubiese llegado hasta aquí.*

## Resumen

La Inteligencia Artificial (IA) es una rama de la ciencia de la computación cuyo principal objetivo es llevar a la computadora las amplias capacidades del pensamiento humano. Dentro del campo de la IA juegan un papel importante los Sistemas Expertos.

En esta investigación se diseña e implementa un sistema experto con el objetivo de contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio de los estudiantes de 10mo grado del IPU "Nieves Morejón López". Para ello se utilizó el lenguaje de programación lógica Prolog, el Netbeans 6.9.1 para la interfaz de usuario y las reglas de producción como forma de representar el conocimiento. La metodología empleada es la propuesta por el Dr.C Mateo Lezcano en su libro: Prolog y los Sistemas Expertos.

El sistema fue diseñado para contribuir a que los estudiantes de 10mo grado sean capaces de reconocer y relacionar los elementos de la Geometría Plana y del Espacio, así como ejercitar los contenidos recibidos en clases.

## Abstract

The Artificial Intelligence (IA) is a branch of the Computation Science whose main objective is to carry out the computer the spacious capacities of the human thought. Inside the field of the IA the Expert Systems play an important paper.

In this research is design and implanted an Expert System with the objective of contribute in the process of teaching and learning of the Plane Geometry and the Space of the ten grade students in Nieves Morejón López school. For all this the logic programation language was used, the Netbeans 6.9.1 to user interfaz and the production rules as form to represent the knowledge. The methodology used is the proposition of Dr.C Mateo Lezcano Brito in his book Prolog and Expert Systems.

The system was design to contribute that the students from the grade be able to recognize and relate the elements of the Plane Geometry and the Space and how to exercise the contents received in classes.

## Contenido

Introducción .....	1
Axiomas, definiciones y teoremas .....	4
Capítulo 1: La Inteligencia Artificial y los sistemas expertos.....	8
1.1    Proceso de Enseñanza _ Aprendizaje de la Matemática .....	8
1.1.1    Surgimiento de la matemática .....	8
1.1.2    Ramas de estudio de las matemáticas. ....	10
1.1.3    Proceso de Enseñanza – Aprendizaje de la geometría en la enseñanza preuniversitaria .....	10
1.1.4    Particularidades del Proceso de enseñanza _ aprendizaje de la Geometría en el Instituto Preuniversitario Urbano “Nieves Morejón López”.....	12
1.2    Los Sistemas Expertos.....	13
1.2.1    Sistemas expertos en la enseñanza.....	14
1.3    Metodologías para desarrollar sistemas expertos .....	16
1.3.1    Metodología de Prototipo .....	17
1.3.2    Metodología Orientada a Objetos .....	19
1.3.3    Metodología descrita por Mateo Lezcano Brito .....	21
1.4    Lenguajes de programación utilizados en la realización de este trabajo.....	23
1.5    Herramientas de desarrollo y tecnologías utilizadas en esta investigación...	25
1.6    Conclusiones .....	26
Capítulo 2: Análisis, diseño e implementación del sistema experto para la geometría plana y del espacio .....	28
2.1 Identificación.....	28
2.2 Establecer conceptos.....	32
2.3 Formalización .....	32
2.3.1 Clasificación del conocimiento .....	32
2.4 Implementación.....	33
2.4.1 Formalización de los casos.....	33
2.4.2 Diseño de los módulos del sistema .....	35
2.4.3 Diagrama de Casos de Uso del sistema.....	35
2.4.4 Descripción de los Casos de Uso del sistema.....	36
2.5 Conclusiones .....	37
Capítulo 3: Descripción de SEPEAGPE.....	38
3.1 Implementación de la interfaz de usuario .....	38
3.1.1 JTextArea.....	39

3.1.3 El uso de iconos, botones y etiquetas.....	40
3.1.4 Otros contenedores.....	40
3.1.5 Barra de menú para un JFrame .....	41
3.1.6 JList.....	42
3.2 Aspectos a tener en cuenta para la interacción de SWI-Prolog y Java .....	42
3.3 Requerimientos del sistema. ....	44
3.3.1 Facilidades que brinda al usuario. ....	44
3.3.2 Evento action.....	45
3.3.3 Menús.....	45
3.3.4 Estándares en la interfaz de la aplicación. ....	47
3.3.5 Concepción general de la ayuda.....	47
3.3.6 Tratamiento de excepciones.....	47
3.4 Conclusiones .....	48
Conclusiones .....	49
Recomendaciones .....	50
Bibliografía .....	51



## Introducción

A medida que ha ido avanzando la ciencia y la tecnología el hombre ha tenido entre sus principales objetivos, lograr que las máquinas piensen igual que el ser humano. En busca de nuevos métodos de aprendizaje para alcanzar tal comportamiento surgió una nueva rama de la Ciencia de la Computación, la Inteligencia Artificial. Este campo de la Computación es el encargado de comprender y replicar la inteligencia humana, además proporciona un conjunto de técnicas, herramientas y métodos que han demostrado su aplicabilidad en muchos campos desde áreas de propósito general como la percepción o el razonamiento, hasta áreas específicas como ingeniería del conocimiento, planificación, videojuegos, entre otras.

Se define la Inteligencia Artificial como aquella inteligencia exhibida por artefactos creados por humanos. A menudo se aplica hipotéticamente a los computadores. El nombre se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas; trata de conseguir que los ordenadores simulen en cierta manera la inteligencia humana. Se acude a sus técnicas cuando es necesario incorporar en un sistema informático, conocimiento o características propias del ser humano. (Inteligencia Artificial).

Uno de los principales objetivos de los investigadores actuales en esta rama de la computación es la reproducción automática del razonamiento humano, donde juegan un papel importante los Sistemas Expertos. Los cuales son una expresión de los Sistemas Basados en el Conocimiento (SBC), los cuales permiten la creación de máquinas que razonan como el hombre, restringiéndose a un espacio de conocimientos limitados. En teoría pueden razonar siguiendo los pasos que seguiría un experto humano (médico, analista, empresario, etc.) para resolver un problema concreto. Los inicios de los Sistemas Expertos datan de mediados de los años sesenta. Durante esta década los investigadores Alan Newell y Herbert Simon desarrollaron un programa llamado GPS (General Problem Solver; solucionador general de problemas). Podía trabajar con criptoaritmética, con las torres de Hanoi y con otros problemas similares. Lo que no podía hacer el GPS era resolver problemas del mundo real, tales como un diagnóstico médico. Para desarrollar Sistema Expertos del tipo "*clasificación*" y "*selección*", Lezcano Brito recomienda como lenguaje de programación lógica, Prolog. (Brito M. L., 1995). Prolog es un lenguaje de programación hecho para representar y utilizar el conocimiento que se tiene sobre un determinado dominio. Un programa escrito en Prolog consiste en un conjunto de reglas y hechos que expresan relaciones entre los objetos del dominio.

Por otra parte, la matemática (del latín **mathematĭca**), aunque con origen más remoto en un vocablo griego que puede traducirse como “conocimiento”, es la ciencia deductiva que se dedica al estudio de las propiedades de los entes abstractos y de sus relaciones. Esto quiere decir que las matemáticas trabajan con números, símbolos, figuras geométricas, etc.

A partir de axiomas y siguiendo razonamientos lógicos, las matemáticas analizan estructuras, magnitudes y vínculos de los entes abstractos. Esto permite, una vez detectados ciertos patrones, formular conjeturas y establecer definiciones a las que se llegan por deducción.

Las matemáticas trabajan con cantidades (números) pero también con construcciones abstractas no cuantitativas. Su finalidad es práctica, ya que las abstracciones y los razonamientos lógicos pueden aplicarse en modelos que permiten desarrollar cálculos, cuentas y mediciones con correlato físico.

Podría decirse que casi todas las actividades humanas tienen algún tipo de vinculación con las matemáticas. Esos vínculos pueden ser evidentes, como en el caso de la ingeniería, o resultar menos notorios, como en la medicina o la música. Es posible dividir las matemáticas en distintas áreas o campos de estudio. En este sentido puede hablarse de la aritmética (el estudio de los números), el álgebra (el estudio de las estructuras), la estadística (el análisis de datos recolectados) y la geometría (el estudio de los segmentos y las figuras), entre otras. (Definición de matemática, 2013).

La geometría, del griego geo (tierra) y metría (medida), es una rama de la matemática que se ocupa de las propiedades de las figuras geométricas en el plano o el espacio, como son: puntos, rectas, planos, polígonos, poliedros, paralelas, perpendiculares, curvas, superficies, etc. Sus orígenes se remontan a la solución de problemas concretos relativos a medidas y es la justificación teórica de muchos instrumentos, por ejemplo el compás, el teodolito y el pantógrafo. Tiene su aplicación práctica en física, mecánica, cartografía, astronomía, náutica, topografía, balística, etc. También sirve como fundamento teórico a inventos como el sistema de posicionamiento global (en especial cuando se la considera en combinación con el análisis matemático y sobre todo con las ecuaciones diferenciales) y es útil en la preparación de diseños (justificación teórica de la geometría descriptiva, del dibujo técnico e incluso en la fabricación de artesanías).

La geometría es una de las más antiguas ciencias. Inicialmente, constituía un cuerpo de conocimientos prácticos en relación con las longitudes, áreas y volúmenes. En el antiguo Egipto estaba muy desarrollada, según los textos de Heródoto, Estrabón y Diodoro Sículo. Euclides, en el siglo III a. C. configuró la geometría en forma axiomática, tratamiento que estableció una norma a seguir durante muchos siglos: la geometría euclidiana descrita en «Los Elementos». El estudio de la astronomía y la cartografía, tratando de determinar las posiciones de estrellas y planetas en la esfera celeste, sirvió como importante fuente de resolución de problemas geométricos durante más de un milenio. René Descartes desarrolló simultáneamente el álgebra y la geometría, marcando una nueva etapa, donde las figuras geométricas, tales como las curvas planas, podrían ser representadas analíticamente, es decir, con funciones y ecuaciones. La geometría se enriquece con el estudio de la estructura intrínseca de los entes geométricos que analizan Euler y Gauss, que condujo a la creación de la topología y la geometría diferencial.

Axiomas, definiciones y teoremas

La geometría se propone ir más allá de lo alcanzado por la intuición. Por ello, es necesario un método riguroso, sin errores; para conseguirlo se han utilizado históricamente los sistemas axiomáticos. El primer sistema axiomático lo establece Euclides, aunque era incompleto. David Hilbert propuso a principios del siglo XX otro sistema axiomático, éste ya completo. Como en todo sistema formal, las definiciones, no sólo pretenden describir las propiedades de los objetos, o sus relaciones. Cuando se axiomatiza algo, los objetos se convierten en entes abstractos ideales y sus relaciones se denominan modelos. Esto significa que las palabras "punto", "recta" y "plano" deben de perder todo significado material. Cualquier conjunto de objetos que verifique las definiciones y los tipos de geometría. (Dinarte, 2011).

Todo esto evidencia que para los estudiantes es importante tener conocimiento y habilidades geométricas, porque les permite estimar y determinar cantidades de magnitudes en situaciones prácticas o de otras áreas del conocimiento o la técnica, les posibilita esbozar o construir figuras y cuerpos geométricos que cumplan determinadas condiciones y los ayuda a descubrir y demostrar nuevas propiedades y relaciones de gran utilidad para interpretar y crear modelos y representaciones de objetos y situaciones de la realidad, contribuyendo con ello también al desarrollo de sus capacidades intelectuales.

En el IPU “Nieves Morejón López” de Cabaiguán, los estudiantes de 10mo grado, como parte de la asignatura matemática, reciben contenidos relacionados con la Geometría Plana y del Espacio. Los resultados de las pruebas finales del 10mo grado, la Prueba Ministerial del 12 grado y Prueba de Ingreso a la Educación Superior, evidencian que existen dificultades en el aprendizaje de estos temas. Entrevistas a profesores y alumnos han permitido identificar las siguientes causas: la cantidad de libros de texto existentes es insuficiente y muchos se encuentran en mal estado; no abundan medios de enseñanza (cuerpos y figuras geométricas) apropiados, los instrumentos de trabajo de los maestros, por ejemplo, el compás de pizarra, no se encuentran en buenas condiciones, lo que trae como consecuencia que a los estudiantes se les hace difícil reconocer y relacionar los elementos; tanto en las figuras planas como en los cuerpos geométricos; así como ejercitar los contenidos recibidos en clases, apropiarse de los conceptos, dominar todas las propiedades de figuras planas, fórmulas del área y perímetro, fórmulas de áreas y volúmenes de los distintos cuerpos geométricos.

Todo lo planteado anteriormente nos conduce al siguiente **problema de investigación:**

¿Cómo contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio de los estudiantes de 10mo grado en el IPU “Nieves Morejón López” de Cabaiguán?

Teniendo en cuenta lo expuesto anteriormente, para dar solución al problema planteado se define como **objetivo general** de este trabajo: Desarrollar un Sistema Experto para contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio de los estudiantes de 10mo grado en el IPU “Nieves Morejón López” de Cabaiguán.

Del objetivo general se plantean las siguientes **Preguntas de Investigación**:

1- ¿Cuáles son los fundamentos teóricos y metodológicos que permitan desarrollar una herramienta informática para contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio?

2- ¿Cómo diseñar un Sistema Experto, basado en la metodología del Dr. Mateo Lezcano para contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio?

3- ¿Cómo implementar un Sistema Experto para contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio de los estudiantes de 10mo grado en el IPU “Nieves Morejón López” de Cabaiguán?

Para responder a las preguntas científicas se plantean las **Tareas de Investigación**:

1- Determinación de los fundamentos teóricos y metodológicos que permitan desarrollar una herramienta informática para contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio.

2- Diseño de un Sistema Experto utilizando la metodología seleccionada para contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio.

3- Implementación de un Sistema Experto para contribuir al Proceso de Enseñanza \_ Aprendizaje de la Geometría Plana y del Espacio de los estudiantes de 10mo grado en el IPU “Nieves Morejón López” de Cabaiguán.

El presente trabajo se estructura de la siguiente forma: introducción, tres capítulos, conclusiones, recomendaciones y bibliografía.

En la **Introducción** se expresa los aspectos más generales de la investigación del diseño teórico y metodológico.

En el **Capítulo 1** se trata el tema de la Inteligencia Artificial y la relación que tienen los Sistemas Expertos con esta rama de la ciencia de la computación. También se abordan elementos de la Matemática y de la rama Geometría Plana y del Espacio.

En el **Capítulo 2** se describen aspectos del dominio de la Geometría Plana y del Espacio y los conceptos descriptivos de sus propiedades, las relaciones que se establecen entre ellos, se organiza el conocimiento y posteriormente se pasa a su formalización e implementación.

En el **Capítulo 3** se realiza la descripción del sistema. Se describe el funcionamiento del sistema.

## Capítulo 1: La Inteligencia Artificial y los sistemas expertos.

Son diversos los campos de aplicación de las técnicas de la IA; entre ellos se destacan: los juegos, el procesamiento de lenguaje natural, la robótica, la educación, la programación automática, el reconocimiento de modelos y los sistemas expertos.

A partir de los años 90 y con el desarrollo de la informática, se produce un amplio desarrollo en el campo de los sistemas expertos, pudiéndose afirmar que estos se han convertido en una herramienta habitual en determinadas empresas en la actualidad. (Castillo, 1999).. En el campo de la enseñanza también se han utilizados de manera exitosa.

### 1.1 Proceso de Enseñanza \_ Aprendizaje de la Matemática.

La **educación matemática** es un término que se refiere tanto al aprendizaje, como la práctica y enseñanza de las matemáticas así como a un campo de la investigación académica sobre esta práctica. Los investigadores en educación matemática en primera instancia, cuestionan las herramientas, métodos y enfoques que faciliten la práctica y/o el estudio de la práctica.

De manera más crítica la educación es más que un simple término, como lo expresa el ilustre pedagogo Rafael Florez Ochoa : " es el proceso social e intersubjetivo mediante el cual cada sociedad asimila a sus nuevos miembros según sus propias reglas, valores, pautas, ideologías, tradiciones, prácticas, proyectos y saberes compartidos por la mayoría de la sociedad. Más modernamente la educación no solo socializa a los individuos sino que también rescata en ellos lo más valioso, aptitudes creativas e innovadoras, los humaniza y potencia como personas". Si además le agregamos el significado de matemática, expresada por el ilustre matemático Bruno D Amore, "son una construcción humana que se utiliza con fines técnicos para la modelización de nuestro entorno y de aplicación en la resolución de problemas prácticos", la **Educación Matemática** se torna compleja en sí misma. (Matemáticas).

#### 1.1.1 Surgimiento de la matemática

La evolución de la matemática puede ser considerada como el resultado de un incremento de la capacidad de abstracción del hombre o como una expansión de la materia estudiada. Los primeros conceptos abstractos utilizados por el hombre,

aunque también por muchos animales, fueron probablemente los Números. Esta noción nació de la necesidad de contar los objetos que nos rodeaban.

Desde el comienzo de la Historia, las principales disciplinas matemáticas surgieron de la necesidad del hombre de hacer cálculos con el fin de controlar los Impuestos y el Comercio, comprender las relaciones entre los números, la medición de terrenos y la predicción de los eventos astronómicos. Estas necesidades están estrechamente relacionadas con las principales propiedades que estudian las matemáticas la cantidad, la estructura, el espacio y el cambio. Desde entonces, las matemáticas han tenido un profuso desarrollo y se ha producido una fructífera interacción entre las matemáticas y la Ciencia, en beneficio de ambas. Diversos descubrimientos matemáticos se han sucedido a lo largo de la historia y se continúan produciendo en la actualidad.

Además de saber contar los objetos físicos, los hombres prehistóricos también sabían cómo contar *cantidades abstractas* como el Tiempo (Días, estaciones, Años, etc.) Asimismo empezaron a dominar la Aritmética elemental (Suma, Resta, Multiplicación y División). Los siguientes avances requirieron la Escritura o algún otro sistema para registrar los números, tales como los tallies o las cuerdas anudadas denominadas Quipu, que eran utilizadas por los Incas para almacenar datos numéricos. Los sistemas de numeración han sido muchos y diversos. Los primeros escritos conocidos que contienen números fueron creados por los egipcios en el Imperio Medio, entre ellos se encuentra el Papiro de Ahmes. La cultura del valle del Indo desarrolló el moderno Sistema decimal, junto con el concepto de Cero.

Los antiguos babilonios utilizaban el Sistema sexagesimal, escala matemática que tiene por Base el número Sesenta. De este sistema la humanidad heredó la división actual del Tiempo: el día en veinticuatro horas o en dos períodos de doce horas cada uno, la Hora en sesenta minutos y el Minuto en sesenta segundos. Los árabes proporcionaron a la cultura europea su Sistema de numeración, que reemplazó a la numeración romana. Este sistema prácticamente no se conocía en Europa antes de que el matemático Leonardo Fibonacci lo introdujera en 1202 en su obra Liber abbaci (Libro del ábaco). En un principio los europeos tardaron en reaccionar, pero hacia finales de la Edad Media habían aceptado el nuevo sistema numérico, cuya sencillez estimuló y alentó el progreso de la Ciencia.

Los Mayas desarrollaron una avanzada Civilización precolombina, con avances notables en la matemática, empleando el concepto del Cero, y en la astronomía, calculando con bastante precisión los ciclos celestes. (Matemáticas)

### **1.1.2 Ramas de estudio de las matemáticas.**

Las numerosas ramas de la matemática están muy interrelacionadas. En una subdivisión amplia de las matemáticas, se distinguen cuatro objetos de estudio básicos: la cantidad, la estructura, el espacio y el cambio. En este trabajo nos centraremos en el espacio.

El estudio del espacio origina la Geometría, primero la Geometría euclídea y luego la Trigonometría. En su faceta avanzada el surgimiento de la topología da la necesaria y correcta manera de pensar acerca de las nociones de cercanía y continuidad de nuestras concepciones espaciales. (Matemáticas)

La Geometría (del latín *geometriā*, que proviene del idioma griego γεωμετρία, *geo* tierra y *metría* medida), es una rama de la matemática que se ocupa del estudio de las propiedades de las figuras geométricas en el plano o el espacio, como son: puntos, rectas, planos (paralelas, perpendiculares, curvas, superficies, polígonos, poliedros, etc).

### **1.1.3 Proceso de Enseñanza – Aprendizaje de la geometría en la enseñanza preuniversitaria**

El proceso de enseñanza consiste, fundamentalmente, en un conjunto de transformaciones sistemáticas de los fenómenos en general, sometidos estos a una serie de cambios graduales cuyas etapas se producen y suceden en orden ascendente, de aquí que se le deba considerar como un proceso progresivo y en constante movimiento, con un desarrollo dinámico en su transformación continua. (Castellano, 2002).

El aprendizaje es un proceso de adquisición de conocimientos, habilidades, valores, actitudes y de una nueva conducta en la que se influyen hábitos y comportamientos de un individuo, mediante la experiencia a consecuencia de su interacción con el medio que lo rodea (Sandoval, 2006).

Sistema de conocimientos de la geometría en el preuniversitario.

Geometría Plana.

La geometría plana es una parte de la geometría que trata de aquellos elementos cuyos puntos están contenidos en un plano. La geometría plana está considerada parte de la geometría euclidiana, pues ésta estudia los elementos geométricos a partir de dos dimensiones.

Los estudiantes de la enseñanza preuniversitaria deben apropiarse de los contenidos referidos a : puntos, rectas, ángulos, polígonos (triángulos, cuadriláteros), circunferencia y círculo, igualdad y semejanza de triángulos, cálculo de área y perímetro, ángulos en la circunferencia. (Elementos y relaciones entre las propiedades y conceptos de las diferentes figuras geométricas).

### Geometría del espacio.

La geometría espacial o geometría del espacio es la rama de la geometría que se ocupa de las propiedades y medidas de las figuras geométricas en el espacio tridimensional o espacio euclidiano. Entre estas figuras, también llamadas sólidos, se encuentran el cono, el cubo, el cilindro, la pirámide, la esfera, el prisma, los poliedros regulares (los sólidos platónicos, convexos, y los sólidos de Kepler-Poinsot, no convexos) y otros poliedros.

Los alumnos de la enseñanza media superior deben conocer los axiomas de la Estereometría (planos, relaciones de posición entre rectas y planos y entre planos). Perpendicular, oblicua, proyección, teorema de las Tres Perpendiculares, cuerpos en el espacio y fórmulas para el cálculo de áreas y volumen.

Habilidades:

En 10mo grado los estudiantes deben dominar las siguientes habilidades en la Geometría Plana y del Espacio

Identificar, comprender, explicar, argumentar, demostrar, calcular, relacionar, ejemplificar, clasificar, definir, generalizar, valorar, interpretar, entre otras.

Medios con los que se cuenta:

Se dispone con diferentes bibliografías; los libros de texto de cada grado (en algunos casos no están contextualizados) y el software Eureka instalado en las computadoras de las escuelas.

Este software fue realizado específicamente para la asignatura de matemática de forma general, contiene conceptos y algunos ejercicios generales de cada una de sus

ramas. A pesar de su utilidad el sistema no profundiza del todo en el contenido de la Geometría Plana y del Espacio.

#### **1.1.4 Particularidades del Proceso de enseñanza \_ aprendizaje de la Geometría en el Instituto Preuniversitario Urbano “Nieves Morejón López”.**

Esta investigación se desarrolla en el Instituto Preuniversitario Urbano “Nieves Morejón López”, fue fundado el 2 de septiembre de 2010, como parte de la propuesta del Ministerio de Educación, de reunificar la red escolar; el mismo se encuentra ubicado en la provincia de Sancti Spíritus, exactamente en el municipio de Cabaiguán en el Consejo Popular Urbano # 1, aproximadamente en el centro de esta localidad, en la intersección de la carretera central que forman la calle República # 87, dicha carretera Ave. Camilo Cienfuegos y Héctor Castellanos. Se encuentra delimitado por una cerca perimetral y cuenta con una sola entrada, y funcionan, además, alternativamente la FOC y la Sede Universitaria Municipal.

A la entrada de este Instituto y a su derecha, se encuentra un mural que lleva por nombre ¡Viva Cuba! Y que es el primero creado por parte del proyecto muralístico “El nombre de mi calle”, de los promotores culturales Lázaro Valdés Pulluelo y Noel Cabrera Pérez. Se escogió este sitio por la ubicación geográfica y por lo que representa para la historia de la localidad, ya que este centro lleva el nombre de uno de los mártires, que cayera brutalmente asesinado cumpliendo una de la misiones asignadas a él por el Directorio Revolucionario. Su diseño y construcción comienza en octubre de 2010, en saludo a la Jornada por la cultura cubana y se intensifica su trabajo a partir de septiembre de 2011, llegando a feliz término el 20 de marzo de 2012, participando en su inauguración la máxima dirección del Partido, el Gobierno y la Dirección Municipal de Educación de este municipio.

Este centro fue readaptado, como es lógico, a las exigencias de este tipo de enseñanza y cuenta hoy con 18 aulas, 5 laboratorios (2 de Computación, 1 de Física, 1 de Química y 1 de Biología); además tenemos una Biblioteca escolar y locales o espacios habilitados para oficinas u otros usos.

La matrícula actual es de 574 estudiantes, procedentes de las tres secundarias básicas del municipio, 71 trabajadores docentes y 22 no docentes, todos con la capacitación necesaria para la función que realizan.

Entrevistas a profesores han permitido identificar que en la enseñanza de la Geometría existen dificultades que atentan contra el aprendizaje del contenido, entre los cuales podemos señalar:

- No se realiza un trabajo asertivo al operar con los conceptos y propiedades básicas.
- Falta rigor en la Comunicación, utilizando la terminología y simbología matemática.
- Es insuficiente el trabajo encaminado a la utilización de recursos y técnicas para la racionalización del trabajo mental.
- La didáctica no está concebida sobre la base de la enseñanza problémica, de manera que no siempre se aprovechan las potencialidades que ofrece el planteamiento de preguntas típicamente matemáticas y la fundamentación de sus conjeturas.
- No se exige al alumno que explique el proceso seguido en la resolución de ejercicios y problemas, haciendo uso de sus ideas matemáticas y sus correspondientes representaciones, en la medida que fundamentan el por qué de la vía de solución elegida, así como el análisis de otras variantes.
- Es insuficiente el trabajo que se realiza en función de evidenciar, a partir de ejercicios, el uso de la matemática en la vida diaria.
- No se realiza un trabajo consecuente con la identificación de las distintas formas de representar los objetos y las situaciones matemáticas, a partir de las relaciones que se establecen entre ellas.
- Es insuficiente el trabajo que se realiza en función de la generalización de representaciones y de procedimientos.
- No se abordan los contenidos, atendiendo al carácter sistémico de la enseñanza.
- Se refleja formalismo al operar con las definiciones de las diferentes figuras geométricas y sus relaciones, predominando el enfoque memorístico en su tratamiento en la clase de Matemática.

## **1.2 Los Sistemas Expertos**

Los Sistemas Expertos forman parte de la ciencia de la computación y dentro de esta se ubican en la rama de la Inteligencia Artificial. Este término fue usado por primera vez por un doctor de la Universidad de Standford, Edgard A. Feigenbaum. Él estipuló que el poder de resolución de un problema en un programa de computadora viene del

conocimiento de un dominio específico, no solo de las técnicas de programación y el formalismo que contiene. Antes de eso casi todos los investigadores en Inteligencia Artificial habían tratado de desarrollar métodos generales de programación que resolverían grandes problemas conceptuales pero estos programas resolvían pobremente los problemas individuales.

En el desarrollo de los Sistemas Expertos también se ha producido a medida que se ha ido desarrollando la Inteligencia Artificial y los diferentes métodos que se han empleado para su resolución. El desarrollo de lenguajes como LISP y PROLOG condicionaron esa evolución, así como investigaciones en diversos campos relacionados. Los primeros Sistemas Expertos que se desarrollaron en los años 1960 eran capaces de resolver solo problemas basadas en situaciones determinadas, mediante sistema de reglas. Es a partir de los años 1970 cuando se empiezan a resolver problemas basados en situaciones inciertas, basados en mediadas difusas al principio y en redes probabilísticas con posterioridad.

Los Sistemas Expertos son llamados así porque emulan el razonamiento de un experto en un dominio concreto y en ocasiones son usados por éstos. Con los sistemas expertos se busca una mejor calidad y rapidez en las respuestas dando así lugar a una mejora de la productividad del experto. Es una aplicación informática capaz de solucionar un conjunto de problemas que exigen un gran conocimiento sobre un determinado tema. Un sistema experto es un conjunto de programas que, sobre una base de conocimientos, posee información de uno o más expertos en un área específica. Se puede entender como una rama de la inteligencia artificial, donde el poder de resolución de un problema en un programa de computadora viene del conocimiento de un dominio específico. Estos sistemas imitan las actividades de un humano para resolver problemas de distinta índole (no necesariamente tiene que ser de inteligencia artificial). También se dice que un SE se basa en el conocimiento declarativo (hechos sobre objetos, situaciones) y el conocimiento de control (información sobre el seguimiento de una acción). (Sistemas Expertos).

### **1.2.1 Sistemas expertos en la enseñanza**

Entre los años 80 a 85 se produce la revolución de los sistemas expertos. En estos cinco años se crearon diversos, entre ellos el DELTA (K., 1994), pero sin duda alguna, una de las áreas de aplicación de la Inteligencia Artificial (IA) que más se desarrolló fue la esfera educacional, específicamente los sistemas para entrenamiento. Los sistemas expertos constituyen un valioso recurso en el proceso docente, incluidos por

algunos autoras como (Galvis Panqueva, A. 1994) y (Alessi, T. 1985) entre los tipos de sistemas de EAC que utilizan técnicas de IA.

Por otra parte, otros autores describen su uso en la educación vinculada al planeamiento, toma de decisiones, adiestramiento del personal docente y profundización en una determinada área del saber.

Según (Laborda, J. 1986), un sistema experto rinda facilidad en su aplicación didáctica permitiendo: Resolver problemas complicados de manera que su forma de operar sirva de guía para el alumno, organizar el saber disponible sobre la materia, posibilitando su aplicación directa a la solución del problema, preservar el conocimiento para su utilización futura, establecer una comunicación eficiente con el estudiante, captar y presentar en diferentes formas las respuestas que recibe o proporciona, reconocer una extensa gama de errores de razonamiento y proveer conjunto de problemas distintos y graduar su dificultad relativa.

Muchos son los SE en la educación que se han desarrollado con diversos fines. Pudieran citarse, entre ellos, el Sistema Experto para contribuir a desarrollar habilidades para Aprender a Aprender (SEPAA) en la asignatura Cognición y Comportamiento I de la carrera de Psicología. El Sistema Experto para contribuir a un mejor aprendizaje de la Ortografía (SEPAO) en la Universidad de Sancti Spíritus José Martí Pérez. Para la realización de estos dos SE se siguió la metodología planteada por el Dr. C. Mateo Lezcano Brito en su libro Prolog y los Sistemas Expertos, para el diseño de la interfaz gráfica se escogió NetBeans como IDE y Java como superlenguaje. La base de conocimiento fue programada en lenguaje Prolog para su manipulación se utilizó SWI Prolog como intérprete. Fueron desarrollados por estudiantes de Ingeniería Informática de la Universidad de Sancti Spíritus “José Martí Pérez”.

Otro ejemplo es el Sistema para la Enseñanza de Sistemas Expertos (SESE), desarrollado en la Universidad Central de Las Villas (Santa Clara, Cuba), para brindar a los aprendices un ambiente interactivo que permita vivir experiencias directas y relevantes al aprendizaje de conceptos y habilidades de programación de sistemas expertos. (Brito v. P., 1998).

También se encuentra Tutoriales Inteligentes aplicando Sistemas Expertos del autor Nicolás Kemper Valverde, desarrollado en la Universidad Nacional Autónoma de México, con el objetivo de discutir las ventajas del uso de la computadora en la educación y de como las técnicas de inteligencia artificial se pueden aplicar para

desarrollar tutoriales inteligentes, discutir los diferentes factores que deben evaluarse durante la planeación de un tutorial inteligente y por último comprender el proceso de desarrollo de un tutorial inteligente, integrando para ello técnicas de sistemas expertos. (Valverde).

Por otra parte existe un artículo llamado Técnicas de Diseño en Sistemas de Enseñanza Inteligentes, cuyo objetivo es hacer una revisión de las diferentes técnicas utilizadas para aplicar la inteligencia artificial en los sistemas de enseñanza inteligentes. Se analizan las distintas técnicas en torno a los cuatro componentes de una arquitectura tradicional: módulo del dominio, modelo de estudiante, módulo tutorial e interfaz y se explican las distintas formas de control incluyendo un control para sistemas de enseñanza reactivo, además se hace una descripción de distintos sistemas de enseñanza inteligentes. (Ana Lilia Laureano Cruces, 2001).

Otro ejemplo es el Sistema Experto para la recomendación de modelos instruccionales: una propuesta para su desarrollo, que recomienda al docente, modelos de enseñanza como macroestrategias pertinentes a la planificación e implementación de la actividad didáctica. (autores, Sistema Experto para la recomendación de modelos instruccionales: una propuesta para su desarrollo).

Se encuentra también el Sistema Experto en el Proceso de Enseñanza – Aprendizaje de las ciencias exactas, este sistema es objeto de apoyo para el aprendizaje y se respalda en los elementos que contribuyen en el diseño del mismo como la arquitectura y los estándares de educación. (autores, Sistema Experto en el Proceso de Enseñanza-Aprendizaje de las ciencias exactas, 2013)

Además se encuentra Enseñanza de las Matemáticas por medio de Sistemas Tutoriales, el cual plantea que el alumno debe trazar la teoría con la práctica, actividad que raras veces se promueve ya que, en el mejor de los casos, los ejercicios que se le plantean suelen ser ajenos a su medio, lo que le dificulta la búsqueda de soluciones. (autores, Enseñanza de las Matemáticas por medio de Sistemas Tutoriales).

### **1.3 Metodologías para desarrollar sistemas expertos**

La creación de las bases de conocimiento es un complejo y largo período de adquisición de información, llevado a cabo por el ingeniero del conocimiento; quien, partiendo de los juicios y opiniones de diferentes expertos, debe codificar la información obtenida utilizando un lenguaje formal de representación del conocimiento.

La culminación exitosa de este trabajo requiere un consenso en cuanto al criterio de cada uno de los expertos, eliminando cualquier discrepancia que pudiera haber surgido en el proceso de refinamiento con respecto a un tema determinado (lo cual ocurre a menudo). Si esto se logra la base de conocimiento resultante contará con una biblioteca de información individual.

Al diseñar una base de conocimiento, se está construyendo algo como una teoría sobre un determinado dominio o que al menos aspira a llegar a serlo, en este sentido deben prevalecer en su diseño los aspectos conceptuales. La información debe estar organizada en forma armónica y debe ser, ante todo, lo más exacta posible. Sin embargo, el conocimiento de los expertos es a menudo incompleto e inconsistente, por lo que el proceso puede tomarse bastante engorroso y en algunos casos, el resultado pudiera no ser lo suficientemente exacto. (Brito M. L., 1995).

### **1.3.1 Metodología de Prototipo**

Si bien algunos autores consideran que esto es parte del ciclo de vida clásico, es también posible verlo como un método independiente.

Una definición de un prototipo en software podría ser:

“... es un modelo de comportamiento del sistema que puede ser usado para entenderlo completamente o ciertos aspectos de él y así clarificar los requerimientos... Un prototipo es una representación de un sistema, aunque no es un sistema completo, posee las características del sistema final o parte de ellas”

Las fases que comprende el método de desarrollo orientado a prototipos serían:

**Investigación preliminar.** Las metas principales de esta fase son: determinar el problema y su ámbito, la importancia y sus efectos potenciales sobre la organización por una parte y, por otro lado, identificar una idea general de la solución para realizar un estudio de factibilidad que determine la factibilidad de una solución software.

**Definición de los requerimientos del sistema.** El objetivo de esta etapa es registrar todos los requerimientos y deseos que los usuarios tienen en relación al proyecto bajo desarrollo. Esta etapa es la más importante de todo el ciclo de vida, es aquí donde el desarrollador determina los requisitos mediante la construcción, demostración y retroalimentaciones del prototipo. Por lo mismo esta etapa será revisada con más detalle luego de esta descripción.

**Diseño técnico.** Durante la construcción del prototipo, el desarrollador ha obviado el diseño detallado. El sistema debe ser entonces rediseñado y documentado según los estándares de la organización y para ayudar a las mantenciones futuras. Esta fase tiene dos etapas: por un lado, la producción de una documentación de diseño que especifica y describe la estructura del software, el control de flujo, las interfaces de usuario y las funciones y, como segunda etapa, la producción de todo lo requerido para promover cualquier mantención futura del software.

**Programación y prueba.** Es donde los cambios identificados en el diseño técnico son implementados y probados para asegurar la corrección y completitud de los mismos con respecto a los requerimientos.

**Operación y mantención.** La instalación del sistema en ambiente de explotación, en este caso, resulta de menor complejidad, ya que se supone que los usuarios han trabajado con el sistema al hacer las pruebas de prototipos. Además la mantención también debería ser una fase menos importante, ya que se supone que el refinamiento del prototipo permitiría una mejor claridad en los requerimientos, por lo cual las mantenciones perfectivas se reducirían. Si eventualmente se requiriese una mantención entonces el proceso de prototipazo es repetido y se definirá un nuevo conjunto de requerimientos.

La fase más importante corresponde a la definición de requerimientos, la cual correspondería a un proceso que busca aproximar las visiones del usuario y del desarrollador mediante sucesivas iteraciones. La definición de requerimientos consiste de cinco etapas entre dos de las cuales se establece un ciclo iterativo:

**Análisis grueso y especificación.** El propósito de esta subfase es desarrollar un diseño básico para el prototipo inicial.

**Diseño y construcción.** El objetivo de esta subfase obtener un prototipo inicial. El desarrollador debe concentrarse en construir un sistema con la máxima funcionalidad, poniendo énfasis en la interface del usuario.

**Evaluación.** Esta etapa tiene dos propósitos: extraer a los usuarios la especificación de los requerimientos adicionales del sistema y verificar que el prototipo desarrollado lo haya sido en concordancia con la definición de requerimientos del sistema. Si los usuarios identifican fallas en el prototipo, entonces el desarrollador simplemente corrige el prototipo antes de la siguiente evaluación. El prototipo es repetidamente modificado y evaluado hasta que todos los requerimientos del sistema han sido

satisfechos. El proceso de evaluación puede ser dividido en cuatro pasos separados: preparación, demostración, uso del prototipo y discusión de comentarios. En esta fase se decide si el prototipo es aceptado o modificado.

**Modificación.** Esto ocurre cuando la definición de requerimientos del sistema es alterada en la subfase de evaluación. El desarrollador entonces debe modificar el prototipo de acuerdo a los comentarios hechos por los usuarios.

**Término.** Una vez que se ha desarrollado el prototipo estable y completo, es necesario ponerse de acuerdo en relación a aspectos de calidad y de representación del sistema.

En el desarrollo de Sistemas Expertos se nos plantean dos importantes riesgos:

- No existen implementaciones similares que puedan servir de orientación al encargado del desarrollo en casi la totalidad de los casos.
- En muchos puntos, los requisitos necesarios están esbozados con muy poca precisión.

El diseño y la especificación requieren una temprana determinación de la interfaz del software y de la funcionalidad de los componentes. Durante el desarrollo, resulta apropiado empezar con implementaciones tipo test para encontrar el camino hacia una solución definitiva y para hacerlas coincidir con las necesidades del usuario.

Un método efectivo es la implementación de un prototipo de Sistema Experto que permita llevar a cabo las funciones más importantes de éste, aunque con un esfuerzo de desarrollo considerablemente inferior al de una implementación convencional. Este proceder se define bajo el nombre de "Rapid Prototyping". Para Sistemas Expertos, el "Rapid Prototyping" es el procedimiento más adecuado, pues posibilita una rápida reacción a los deseos en constante cambio tanto por parte de los expertos como por parte del usuario.

### **1.3.2 Metodología Orientada a Objetos**

La programación orientada a objetos (OOP en adelante) es un tipo de programación que provee una manera de modularizar programas estableciendo áreas de memorias particionadas para datos y procedimientos, que pueden ser usadas como plantillas para crear copias de tales módulos conforme se requieran.

Cada área de memoria a la que se refiere en el párrafo anterior es conocida como clase, mientras que las copias creadas a partir de la clase son llamadas objetos. La OOP tiene varias propiedades, entre las cuales destacan:

- **Abstracción:** Permite enfocarse en la solución general del problema, sin preocuparse de los detalles.
- **Encapsulación:** Es el concepto de que un objeto debería tener separada su interfaz de su implementación. Es decir, un objeto es visto como una “caja negra”.
- **Polimorfismo:** Significa que un objeto se puede comportar de diversas maneras, dependiendo del contexto en el que se encuentre.
- **Herencia:** Significa que se pueden crear clases que hereden el comportamiento de una o más clases padres; y que además añadan su propio comportamiento.

El concepto de encapsulación evita que se pueda modificar el estado interno de un objeto por el código cliente que lo utiliza, es decir solamente el código mismo del objeto puede modificar el estado interno del mismo. Esto se puede aprovechar en la construcción de SE creando clases que correspondan a los componentes de un SE.

Por lo tanto si se utiliza la metodología orientada a objetos a la construcción de SE, se debe crear una clase que contenga el código referente a la base de conocimiento.

Si dicha base de conocimiento es un conjunto de objetos con sus propiedades, entonces dicha clase será una clase contenedor de objetos de otra clase, los cuales contendrán el nombre del objeto y una lista de atributos.

De manera similar, el motor de inferencia se implementa como una clase, la cual contiene encapsulada toda la lógica que hace que funcione el SE, es decir si se implementa usando encadenamiento hacia delante, encadenamiento hacia atrás o reglas de producción, dicha implementación estará contenida dentro de la clase.

Esta metodología tiene diversas ventajas, entre las cuales destaca el hecho que cualquier modificación o mantenimiento que se le quiera realizar a un determinado componente no afectará al otro y viceversa. Tello propone crear especializaciones o jerarquía de clases (usando herencia) que incluyan métodos que permitan a los objetos modificarse a sí mismos, logrando con esto que un SE aprenda por sí mismo, modificando su base de conocimiento sin modificar los demás componentes. Otra

ventaja de la OOP es que mediante la herencia, las clases hijas heredarán el comportamiento de clases padres, evitando con esto, tener que incluir código redundante en las especializaciones. (Cando, 2012).

### **1.3.3 Metodología descrita por Mateo Lezcano Brito**

La creación de las bases de conocimiento es un complejo y largo periodo de adquisición de información, llevado a cabo por el ingeniero del conocimiento; quién, partiendo de los juicios y opiniones de diferentes expertos, debe codificar la información obtenida utilizando un lenguaje formal de representación del conocimiento. La culminación exitosa de este trabajo requiere un consenso en cuanto al criterio de cada uno de los expertos, eliminando cualquier discrepancia que pudiera haber surgido en el proceso de refinamiento con respecto a un tema determinado (lo cual ocurre a menudo). Si esto se logra, la base de conocimiento resultante contará con una biblioteca de información mayor que cualquier caudal de información individual.

Cuando se diseña una base de conocimientos, se está construyendo algo como una teoría sobre un determinado dominio o que al menos aspira a serlo, en este sentido deben prevalecer en su diseño los aspectos conceptuales. La información debe estar organizada en forma armónica y debe ser, ante todo, lo más exacta posible. Sin embargo, el conocimientos de los expertos es a menudo incompleto e inconsistente, por lo que el proceso puede tomarse bastante engorroso y en algunos casos, el resultado pudiera no ser lo suficiente exacto. (Brito M. L., 1995)

La base de conocimiento debe ser consecuente, es decir, se debe construir a partir de información precisa, obtenida de la práctica, experiencia y opiniones del experto y que haya sido probada en el medio adecuado con resultados exitosos.

Los procedimientos de la base de conocimiento deben responder a una metodología consistente y correcta, partiendo de la información obtenida en base a la regla anterior.

Se debe proporcionar una estructura adecuada para la socialización con el hombre. Los mensaje y preguntas se deben presentar en forma semejante a las que se usan en el medio real, puesto que las preguntas que carecen de claridad pueden dar lugar a ambigüedades y disgustos por parte del usuario, provocando además ciertas limitaciones al sistema.

Se debe proporcionar un tiempo de respuesta adecuado, de forma que se aumente la productividad del sistema. Esto implica que se eviten, por todos los medios, iteraciones y lazos innecesarios, para mantener de esta forma el tiempo de espera al mínimo.

Debe brindar facilidades para incorporar nuevos conocimientos.

Un sistema experto es tan bueno como lo sea su base de conocimiento, por tanto en su construcción se deben emplear todos los esfuerzos y recursos necesarios, así como todo el tiempo que se disponga. Se deben realizar tantas iteraciones como sean precisas hasta que el sistema alcance su estabilidad.

Existe un método clásico general, que permite la separación de este proceso en varias etapas. El primer paso a seguir cuando se habla de la construcción de un SE es el estudio del dominio, con vista a determinar si es realmente propicia la construcción del SE o no.

Este análisis se puede hacer en base a los siguientes criterios:

- Inexistencia de una solución algorítmica para el problema.
- Se trata de problemas relativamente estáticos, comparados con el tiempo requerido para analizarlos.
- Las tareas no son muy fáciles de resolver (se requieren años para formar un perito y además son muy escasos).
- La tarea debe tener interés práctico, produciendo altos beneficios.
- El conocimiento humano pudiera perderse.

Después de hecho este análisis, si se determina que es posible la construcción del SE, se pasa a las diferentes etapas como son:

**Identificación del problema:** En esta etapa se determina, básicamente, cuál es el problema que se quiere resolver y sus características, así como quienes van a participar en el levantamiento de la base de conocimiento y el papel de cada cual en ese proceso, entre otras cosas. Es más bien un periodo de familiarización, donde tiene lugar la relación inicial entre el experto y el ingeniero del conocimiento. En una entrevista inicial (y otras sucesivas si es necesario) el experto debe lograr que el ingeniero de conocimiento se lleve una idea general del dominio que se quiere modelar, haciendo una caracterización informal del problema y mostrando algunas descripciones de problemas típicos y los posibles pasos para su solución.

**Establecer conceptos:** Se definen los conceptos para la representación del conocimiento. El experto y el ingeniero del conocimiento determinan los aspectos

claves del problema, las relaciones entre ellos y sus características con el objetivo de llevar a cabo la descripción del proceso de solución del problema.

**Formalización:** En esta etapa se formalizan los conceptos claves y subproblemas que estaban aislados durante la conceptualización. Se diseñan las estructuras para organizar el conocimiento. Después de un análisis intensivo, por parte del ingeniero de conocimiento, de los diferentes medios de representación con que se cuenta, se determina cuál se adapta mejor a las condiciones del problema, estableciendo un lenguaje formal que incorpore los conceptos formalizados del tema objeto de representación y describa a la vez el mecanismo de solución.

La definición de los conceptos y relaciones derivadas (conocimiento inducido) y la creación de reglas que los relacionan con el conocimiento explícito, marca el paso del plano conceptual al formal.

En esta etapa el ingeniero del conocimiento juega un papel más activo, determinando: la existencia de datos redundantes, si hay incertidumbre asociada a los datos, si los datos son consistentes y completos para resolver el problema y además si la interpretación lógica de los datos depende de su orden de ocurrencia en el tiempo.

Cumpliendo correctamente todas las etapas se formalizan los conceptos claves y subproblemas y se pasa a la implementación.

#### **1.4 Lenguajes de programación utilizados en la realización de este trabajo**

Proceso Unificado de Software (UML).

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

## Java

Java es un lenguaje de programación orientado a objetos desarrollado por la empresa Sun Microsystems en 1995 y que se ha extendido ampliamente en Internet. Es un lenguaje de alto nivel y propósito general, similar a C++, con marcadas características de seguridad y transportabilidad (Camallea & Abalo, 2005). Uno de los rasgos más importantes es que los programas “ejecutables”, creados por el compilador de Java, son independientes de la arquitectura. Se ejecutan indistintamente en una gran variedad de equipos con diferentes procesadores y sistemas operativos. Pero además Java se caracteriza por: ser un lenguaje intrínsecamente orientado a objetos, funcionar perfectamente en red, aprovechar características de la mayoría de los lenguajes modernos evitando sus inconvenientes. En particular los del C++, tienen una gran funcionalidad gracias a sus librerías (clases), no tienen punteros manejables por el programador, aunque los maneja interna y transparentemente, el manejo de la memoria no es un problema, la gestiona el propio lenguaje y no el programador y generar aplicaciones con pocos errores posibles.

Java es uno de los lenguajes de programación más utilizados en el desarrollo de aplicaciones de código abierto y posee una variada bibliografía para aquellos programadores interesados en el aprendizaje de este lenguaje de programación Java para la realización de este proyecto.

## Prolog

Prolog es un lenguaje de programación para manipular objetos y las relaciones entre ellos y se clasifica como un lenguaje de programación lógica debido a que se basa en la prueba de teoremas, a partir de una base de datos interna formada por reglas escritas en la forma de cláusulas de Horn, donde se aplica el principio de resolución y de unificación. Se dice que Prolog es adecuado para buscar soluciones de problemas que no sabemos resolver (Brito M. L., 1995).

Prolog se conoce como un lenguaje declaratorio. Esto significa que dados los hechos necesarios y reglas, Prolog usará el razonamiento deductivo para resolver sus problemas de la programación. Esto está en contraste con los lenguajes de la computadora tradicionales, como el C, Basic y Pascal que son los lenguajes procedurales.

En un lenguaje procedural, el programador debe proporcionar instrucciones que dicen exactamente a la computadora cómo resolver un problema dado paso a paso. En otros términos, el programador debe saber resolver el problema antes de que la

computadora pueda hacerlo, por otro lado, sólo necesita proporcionar una descripción del sistema y aterrizar las reglas para resolverlo. De allí, Prolog determina cómo encontrar una solución.

Entre sus beneficios podemos destacar:

- 1- Permite crear programas de la Inteligencia Artificial (IA) mucho más rápido y más fácil.
- 2- Es ideal para implementar sistemas expertos y procesamiento del lenguaje natural.
- 3- Los mecanismos de inferencias y los procedimientos son parte de él (built-in).

Este lenguaje de programación lógica es considerado apropiado para la IA por las facilidades que ofrece para la representación del conocimiento, es útil para la realización de sistemas de enseñanza inteligentes. El grado de generalización de las experiencias alcanzadas en la elaboración de entrenadores resulta mayor cuando se utiliza Prolog.

ADA PD Prolog es un intérprete antiguo (1986) y lento para MS-DOS de Autómata Design Associates.

Arity/Prolog32 es un entorno completo de programación en Prolog para Windows de 32 bits que incluye un verdadero compilador, editor, depurador, intérprete y ayuda.

B-Prolog es un sistema CLP (Constraint Logia Programming) que ejecuta programas Prolog. Hay versiones para Windows y varios Unix, incluyendo Linux. Se distribuye también el código fuente.

AMZI! Prolog posee una interfaz simple para generar aplicaciones de consola. Pero por supuesto que esto no es suficiente cuando es necesario utilizar Prolog para aplicaciones profesionales que requieren una interfaz más amigable y accesible para todo tipo de usuarios; (Tito, 2006).

### **1.5 Herramientas de desarrollo y tecnologías utilizadas en esta investigación**

En el proceso de selección de las herramientas a utilizar para la implementación del software que respalda esta investigación, se realizó un levantamiento del estado actual de las tecnologías que actualmente se usan en el desarrollo de sistemas similares.

## SWI-Prolog

SWI-Prolog trabaja en un entorno de Software Libre, licenciado bajo la Lesser GNU Public License y posee un conjunto de herramientas de gráficos XPCE, su desarrollo se inició en 1987 y ha sido impulsada por las necesidades de aplicaciones en el mundo real. Actualmente SWI-Prolog se utiliza en la investigación, para aplicaciones comerciales y en la educación (Martin, 2010).

SWI-Prolog es una de las implementaciones de Prolog fácil y amigable que se acopla muy bien con Windows. Una de las facilidades que esta versión ofrece es que es ampliamente compatible con muchos lenguajes de programación, pues ofrece interfaces de comunicación con Visual Basic, C++, MS Office, Delphi, Java y otros.

## Netbeans

Netbeans es un entorno de desarrollo integrado (IDE) de gran éxito con gran base de usuarios, este proyecto de código abierto fue creado por Sun Microsystems en junio de 2000 el cual continúa siendo el patrocinador principal de los proyectos (Netbeans, 2009).

La Plataforma Netbeans es una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones (Netbeans, 2009).

## Visual Paradigm

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. También proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

### **1.6 Conclusiones**

En este capítulo se recogen aspectos de la Inteligencia Artificial, los Sistemas Expertos y sus aplicaciones; además el surgimiento de la matemática, sus ramas de

estudio, especialmente la geometría. Se precisa la conveniencia de la elección del lenguaje Prolog, para la confección de la base de conocimiento y de Netbeans para modelar el diseño.

## **Capítulo 2: Análisis, diseño e implementación del sistema experto para la geometría plana y del espacio**

La Metodología que se utilizó en el desarrollo de esta investigación fue la descrita por el Dr. Mateo Lezcano Brito en su libro Prolog y los Sistemas Expertos, la cual plantea que primeramente se identifican los elementos del dominio y los conceptos descriptivos de sus propiedades, se define las relaciones que se establecen entre ellos, en base a las cuales se organiza el conocimiento y posteriormente se pasa a su formalización e implementación.

### **2.1 Identificación**

Los métodos de construcción de sistemas expertos parten, en primer lugar, de la identificación de los elementos del dominio y los conceptos descriptivos de sus propiedades, definiendo además las relaciones que se establecen entre ellos, en base a las cuales se organiza el conocimiento (Martin, 2010).

En esta etapa se determinó, básicamente, el problema a resolver, el cual se basa en cómo contribuir a la clasificación de las Figuras de la Geometría Plana y del Espacio, mediante un Sistema Experto y como valor agregado se determinó facilitar al docente un medio para la enseñanza de la clasificación de las Figuras de la Geometría Plana y del Espacio. Se acordó quiénes participarían en el levantamiento de la base de conocimiento y el papel de cada cual en ese proceso. También se determinaron los elementos del dominio con sus conceptos descriptivos.

Los participantes en el levantamiento de la base de conocimiento fueron:

- MSc. Zulema Pérez Pérez experto en Geometría.
- Geidy Carera Pérez.

De forma conjunta, se determinaron los elementos del dominio con sus conceptos descriptivos, los que se relacionan seguidamente:

Elementos del dominio:

## 1. Figuras de la Geometría Plana

- Polígono: Es al región del plano limitada por una línea poligonal cerrada de cualquier número de lados.
- Polígonos regulares: Son aquellos que tienen todos sus lados y sus ángulos iguales.
- Cuadrilátero: Es un polígono de cuatro lados.
- Cuadriláteros convexos: Es la porción del plano limitada por una línea poligonal cerrada y convexa de cuatro lados.
- Trapezoide: Es el cuadrilátero convexo que no tiene lados paralelos.
- Trapezoide simétrico: Es el trapezoide que tiene dos pares de lados consecutivos iguales.
- Trapezio: Son los cuadriláteros convexos de cuatro lados y un par de lados paralelos llamados bases.
- Paralelogramo: Es el trapezio que tiene sus pares de lados paralelos.
- Rombo: Es el paralelogramo que tiene todos sus lados iguales.
- Rectángulo: Es el paralelogramo que tiene sus cuatro ángulos rectos.
- Cuadrado: Es el paralelogramo que tiene todos sus lados iguales y sus ángulos rectos.
- Triángulo: Es un polígono de tres lados.
- Circunferencia: Es el conjunto de todos los puntos del plano que distan de un punto fijo llamado centro.
- Círculo: Es el conjunto formado por todos los puntos de una circunferencia y sus puntos interiores.

## 2. Figuras de la Geometría del Espacio.

- Ortoedro: Cuerpo cuyas aristas laterales son perpendiculares a los planos de las bases y sus bases son rectángulos.
- Cubo: Es el ortoedro que tiene iguales todas sus aristas y sus seis caras son cuadrados iguales.
- Prisma: Cuerpo limitado por varios paralelogramos y dos polígonos iguales cuyos planos son paralelos.
- Pirámide: Cuerpo que una de sus caras es un polígono cualquiera, la cual es la base de la pirámide, y las otras caras son triángulos que tienen un vértice común que es el de la pirámide.
- Cilindro circular recto: Cuerpo que se obtiene por la rotación de un rectángulo alrededor de uno de sus lados.
- Cono circular recto: Cuerpo que se obtiene por la rotación de un triángulo rectángulo alrededor de uno de sus catetos.
- Esfera: Cuerpo que se obtiene por la rotación de un semicírculo alrededor de su diámetro. El centro, el radio y el diámetro del semicírculo son de la esfera respectivamente.

Los objetivos relevantes e irrelevantes de la investigación son:

Relevantes: Suministrar una herramienta a los estudiantes para el aprendizaje de la Geometría Plana y del Espacio.

Irrelevantes: Suministrar un medio a los docentes para la enseñanza de la Geometría Plana y del Espacio.

Los recursos con que se cuenta para la obtención de información son los siguientes:

Experto en Geometría MsC. Zulema Pérez Pérez.

Libros:

- Complementos de geometría y nociones de cálculo diferencial o integral, Dr. Mario O. Glez.
- Entrénate en la Geometría, MSc. Richard Naredo Castellanos.

Tesis:

- Sistema Experto para la clasificación temprana de Dengue Severo en el Instituto Pedro Kourí (IPK) de la Ciudad de la Habana.
- Sistema Experto para contribuir a desarrollar habilidades para Aprender a Aprender (SEPAA).
- Sistema Experto para contribuir a desarrollar habilidades para Aprender Ortografía (SEPAO).
- Entrevistas a : MSc. Zulema Pérez Pérez

El alcance que tiene, en un inicio, el problema y las diferentes submetas trazadas son:

Alcance:

-En el aprendizaje: Ayuda a los estudiantes a ejercitar los contenidos recibidos en clase.

-En la enseñanza: Facilita a los docentes un medio interactivo que contiene los conceptos, propiedades de las figuras planas, fórmulas del área y perímetro, axiomas de la geometría del espacio, teoremas y fórmulas de área y volumen de los distintos cuerpos geométricos.

Submetas trazadas:

- Estudiar toda la bibliografía disponible.
- Confeccionar la base de conocimiento.
- Confeccionar la interfaz visual.

Se utilizó SWI-Prolog para la confección de la Base de Conocimiento y NetBeans 6.9.1 para el desarrollo de la interfaz visual, por ser un entorno de desarrollo de aplicaciones rápido, seguro, portable, simple y robusto, además de poseer facilidades para interactuar con SWI-Prolog a través de la librería jpl.

## 2.2 Establecer conceptos

En este caso, se estableció que el conocimiento se organizaría atendiendo a los elementos del dominio y los conceptos que lo describen, y en base al elemento del dominio que el usuario desee consultar se realizaría el proceso de razonamiento.

En este período se definieron los conceptos para la representación del conocimiento, se determinaron los aspectos claves del problema, las relaciones entre ellos y sus características con el objetivo de llevar a cabo la descripción del proceso de solución.

Se establecen una serie de elementos esenciales, tales como:

- Los datos de salida son los conceptos, características y propiedades de cada figura plana y cuerpo geométrico.

El conocimiento se organizó atendiendo a los elementos del dominio y los conceptos descriptivos de estos como ya se había planteado anteriormente, y en base al elemento del dominio que el usuario desee consultar es que se realiza el proceso de razonamiento. Por ejemplo, si el usuario decide consultar la opción. Geometría Plana, el sistema tiene varias opciones: Concepto, Área y Perímetro y algunos ejemplos de ejercicios, si el usuario necesita saber algún concepto busca la figura que necesita

## 2.3 Formalización

Corresponde a esta etapa formalizar los conceptos claves identificados en la fase anterior y los subproblemas que estaban aislados durante la conceptualización. Se diseñan las estructuras para organizar el conocimiento.

### 2.3.1 Clasificación del conocimiento

La organización del conocimiento quedó dividida en **dos módulos** con el objetivo de organizar el conocimiento desde el punto de vista lógico.

**Módulo 1:** Nombre: Figuras de la Geometría Plana.

**Módulo 2:** Nombre: Figuras de la Geometría del Espacio.

## **2.4 Implementación**

En esta etapa el ingeniero de conocimiento combina y reorganiza el conocimiento formalizado para hacerlo compatible con las características del flujo de información del problema.

Se demuestra que los métodos de solución seleccionados son los más indicados para la exitosa solución de, al menos, un primer grupo de problemas, comenzando por los más generales y urgentes y pasando posteriormente a un refinamiento del conocimiento.

El proceso de refinamiento consiste en el mejoramiento del conocimiento ya existente y la adición de nuevos fragmentos de conocimiento, además de mejorar la estrategia de búsqueda del sistema, lo que contribuye a una solución más efectiva y confiable.

En la práctica, este proceso no es tan fácil como puede parecer. Usualmente se llevan a cabo múltiples iteraciones, seguidas por una total o parcial reimplantación de la base de conocimiento antes que comience a ser útil el diseño del sistema.

La realización de pruebas al primer prototipo y el análisis de las críticas que se hagan a su funcionamiento servirán como punto de partida para la formalización de una versión más avanzada de la base de conocimiento.

En esta investigación, durante esta fase se llevó a cabo la formalización de las reglas que abarcan todo el conocimiento, obteniéndose un primer prototipo, el que se sometió a períodos de prueba y puesta a punto hasta conseguir una versión más avanzada de la base de conocimiento.

### **2.4.1 Formalización de los casos**

Algunas reglas que abarcan el conocimiento quedaron formalizadas de la siguiente manera:

trapezio: - tiene\_cuatro\_lados,  
          tiene\_par\_de\_lados\_paralelos\_llamados\_bases.

paralelogramo: - tiene\_cuatro\_lados,  
                  tiene\_par\_de\_lados\_llamados\_bases,  
                  tiene\_sus\_pares\_de\_lados\_paralelos.

Las reglas anteriormente descritas son solo una porción de todas las que fueron implementadas en SWI-Prolog, en una versión inicial, luego se hizo una segunda versión modificada en SWI-Prolog nuevamente con la intención de

lograr una interfaz visual más amigable mediante la interacción con Netbeans, utilizando el lenguaje Java y una librería con la que este cuenta que nos facilita la conexión con la Base de Conocimiento.

Se tuvo que realizar un cambio radical en la Base de Conocimiento para poder utilizar las clases con las que cuenta la Librería jpl.

A continuación se muestra un fragmento de cómo quedo el código en la segunda versión:

```
tiene_cuatro_lados(si,'¿Tiene par de lados llamados bases? ', 'tiene_par_de_lados_llamados_bases(-,Y,Z)').
```

```
tiene_cuatro_lados(no,'¿Tiene tres lados? ', 'tiene_tres_lados(-,Y,Z)').
```

```
tiene_par_de_lados_llamados_bases(si,'La figura es un trapecio','-').
```

```
tiene_par_de_lados_llamados_bases(no,'¿Tiene par de lados paralelos? ', 'tiene_par_de_lados_paralelos(-,Y,Z)').
```

```
tiene_tres_lados(si,'La figura es un triangulo','-').
```

```
tiene_tres_lados(no,'¿Es un conjunto de puntos del plano que equidistan de un punto llamado centro? ', 'conjunto_de_puntos(-,Y,Z)').
```

```
tiene_par_de_lados_paralelos(si,'La figura es un paralelogramo','-').
```

```
tiene_par_de_lados_paralelos(no,'¿Tiene sus lados iguales? ', 'tiene_sus_lados_iguales(-,Y,Z)').
```

```
tiene_sus_lados_iguales(si,'La figura es un rombo','-').
```

```
tiene_sus_lados_iguales(no,'¿Tiene sus angulos rectos? ', 'tiene_sus_angulos_rectos(-,Y,Z)').
```

```
tiene_sus_angulos_rectos(si,'La figura es un rectangulo','-').
```

```
tiene_sus_angulos_rectos(no,'La figura es desconocida','-').
```

```
conjunto_de_puntos(si,'La figura es una circunferencia','-').
```

```
conjunto_de_puntos(no,'¿Es un conjunto formado por los puntos de una circunferencia y sus puntos interiores? ', 'conjunto_formado(-,Y,Z)').
```

### 2.4.2 Diseño de los módulos del sistema

Cada módulo posee una base de conocimiento, la cual está dividida en tres partes, un bloque de preguntas, un bloque de reglas y un bloque de explicación; además cuenta con un motor de inferencia y una interfaz del usuario. En la siguiente figura se presenta el diseño.

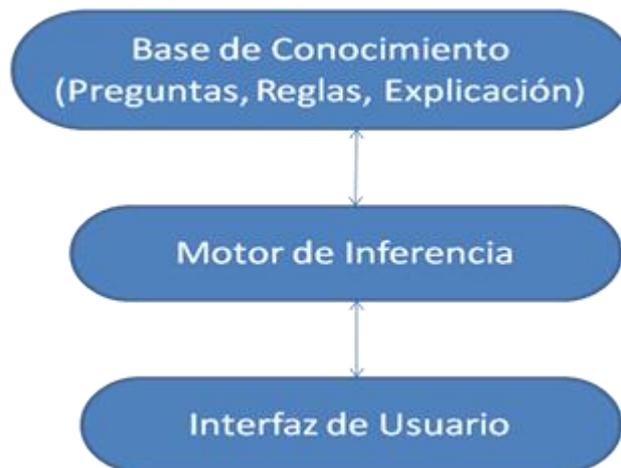


Ilustración 1: Diseño de los módulos del sistema

### 2.4.3 Diagrama de Casos de Uso del sistema

El modelo de casos de uso permitió al desarrollador del sistema y a los clientes que llegaran a un acuerdo sobre las condiciones y posibilidades que debe tener el sistema.

Este diagrama se muestra en la figura.

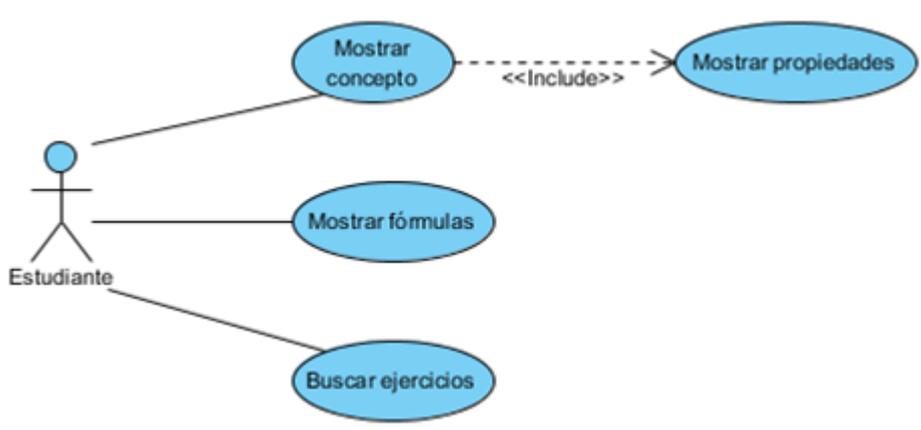


Ilustración 2: Diagrama de Casos de Uso del sistema

#### 2.4.4 Descripción de los Casos de Uso del sistema

El modelado de Casos de Uso es la técnica más simple y que emplean los desarrolladores de software para modelar los requisitos del sistema desde la perspectiva del usuario. El modelo de casos de uso consiste en actores y casos de uso. Los actores representan usuarios y otros sistemas que interactúan con el sistema y los casos de uso representan el comportamiento del sistema, los escenarios que el sistema atraviesa en respuesta a un estímulo desde un actor (System, 2005)

Tabla 1: Descripción del Caso de uso: Mostrar concepto

<b>Caso de uso:</b> Mostrar concepto.
<b>Actores:</b> Estudiante (inicia).
<b>Propósito:</b> Solicitar ayuda al sistema para saber el concepto de alguna figura y sus propiedades.
<b>Resumen:</b> El caso de uso se inicia cuando un estudiante, desea consultar el sistema para saber el concepto de alguna figura y sus propiedades. El sistema le muestra el concepto y la imagen de la figura que desee consultar, además de las propiedades de la misma. El caso de uso finaliza cuando el sistema le muestre al estudiante lo que necesite consultar.
<b>Pre-condiciones:</b> Tener guardado los conceptos de las figuras y cuerpos geométricos.
<b>Pos-condiciones:</b> El sistema muestra el concepto de la figura o cuerpo geométrico solicitado.
<b>Prototipo:</b> Ver anexo o figura del capítulo 3 que muestre un ejemplo de alguna figura.

Tabla 2: Descripción del Caso de uso: Mostrar fórmulas.

<b>Caso de uso:</b> Buscar fórmulas.
<b>Actores:</b> Estudiante (inicia).
<b>Propósito:</b> Solicitar ayuda al sistema para saber el área, perímetro o volumen de alguna figura.
<b>Resumen:</b> El caso de uso se inicia cuando un estudiante, desea consultar el sistema para saber el área, perímetro o volumen de alguna figura. El sistema le muestra las fórmulas y la imagen de la figura que desee consultar. El caso de uso finaliza cuando el sistema le muestre al estudiante lo que necesite consultar.
<b>Pre-condiciones:</b> Tener guardado las fórmulas de las figuras y cuerpos geométricos.
<b>Pos-condiciones:</b> El sistema muestra el concepto de la figura o cuerpo geométrico

solicitado.
<b>Prototipo:</b> Ver anexo o figura del capítulo 3 que muestre un ejemplo de alguna figura.

Tabla 3: Descripción del Caso de uso: Buscar ejercicios.

<b>Caso de uso:</b> Buscar ejercicios.
<b>Actores:</b> Estudiante (inicia).
<b>Propósito:</b> Solicitar ayuda al sistema para saber cómo resolver algún tipo de ejercicio.
<b>Resumen:</b> El caso de uso se inicia cuando un estudiante, desea consultar el sistema para saber cómo resolver algún tipo de ejercicio. El sistema le muestra los pasos para resolver los ejercicios con la imagen de la figura. El caso de uso finaliza cuando el sistema le muestre al estudiante lo que necesite ejercitar.
<b>Pre-condiciones:</b> Tener guardado los ejercicios.
<b>Pos-condiciones:</b> El sistema muestra los ejercicios.
<b>Prototipo:</b> Ver anexo o figura del capítulo 3 que muestre un ejemplo de algún ejercicio.

## 2.5 Conclusiones

En el presente capítulo se identificaron los elementos del dominio y los conceptos descriptivos de sus propiedades; y las relaciones entre ellos. Se formalizaron e implementaron las reglas. Se obtuvieron varios casos de uso y un primer prototipo del sistema, este se desarrolló según la metodología descrita por el Dr. Mateo Lezcano en su libro Prolog y los Sistemas Expertos. También se definieron los casos de usos del sistema.

## Capítulo 3: Descripción de SEPEAGPE

En este capítulo se explican los requerimientos del hardware y software para lograr un rendimiento óptimo del sistema experto, así como las facilidades que brinda el mismo desde el punto de vista del usuario, terminando este capítulo con un manual de usuario para la correcta utilización del software.

### 3.1 Implementación de la interfaz de usuario

Después de la formalización de la base de conocimiento, mediante las reglas de producción y la definición de los requerimientos del sistema se pasó a la implementación de la interfaz del usuario. Para desarrollar la interface visual se utilizó NetBeans 6.9.1 por ser un entorno de desarrollo rápido de aplicaciones y poseer facilidades para interactuar con SWI-Prolog. En SWI-Prolog se genera un fichero .pl, el cual se utiliza para consultar la base de conocimiento.

SWI-Prolog posee una interfaz simple para generar aplicaciones de consola, pero esto no es suficiente cuando necesitamos utilizar Prolog para aplicaciones profesionales que requieren una interfaz más amigable y accesible para todo tipo de usuarios, para solucionar este problema algunas implementaciones incluyen bibliotecas que permiten realizar la interfaz de Prolog con superlenguajes como Java. El uso de SWI-Prolog con Java se facilita por una componente (la clase Query) que implementa las llamadas a la biblioteca, así como el acceso a otros tipos de datos necesarios. La biblioteca **jpl.jar** disponible en SWI-Prolog brinda una gran variedad de procedimientos y funciones para manipulación de datos, predicados y términos de Prolog.

#### Diseño de la interfaz gráfica.

Dentro de los cuatro componentes Swing muy utilizados que crea SwingApplication se encuentra un frame, o ventana principal (JFrame), un panel, llamado (JPanel), varios botones llamados (JButton) y etiquetas que tienen como nombre (JLabel). GeoPE

utiliza para la ventana principal un JÚrame, como se recomienda en este caso y JDialog para el resto de las ventanas.

### 3.1.1 JTextArea

Los JTextArea son utilizados para mostrar extensos textos extraídos de la base de conocimiento.



Figura 1: Uso de JTextArea en SEPEAGPE.

### 3.1.2 RadioButtons

Los RadioButtons son utilizados para seleccionar la opción deseada.

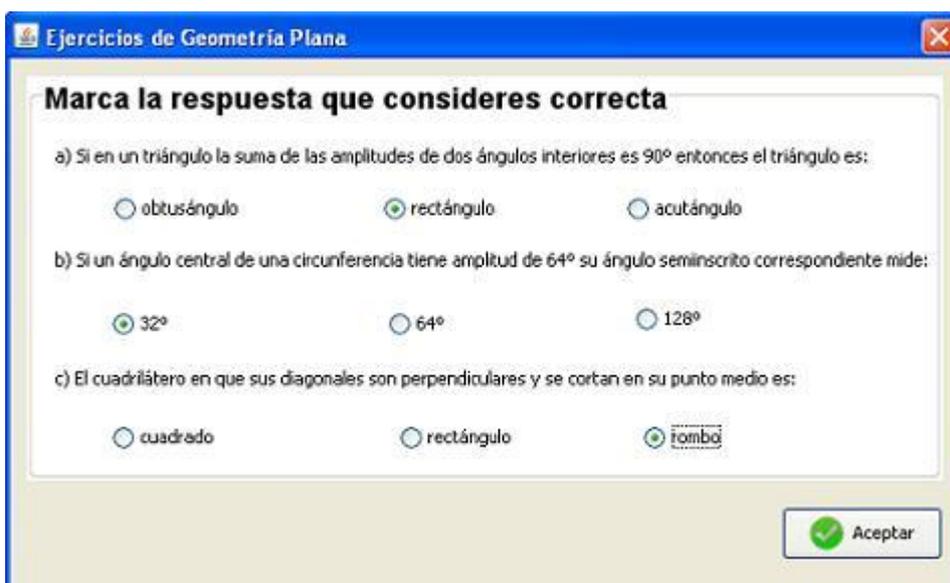


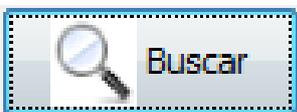
Figura 2: Uso de RadioButtons en SEPEAGPE.

### 3.1.3 El uso de iconos, botones y etiquetas

Algunos componentes Swing, como JLabel y JButton, pueden ser decorados con un icono, una imagen de tamaño fijo. En Swing, un icono es un objeto que se adhiere al interface Icon. Un objeto ImageIcon puede cargar una imagen desde un nombre de fichero, una URL u otra fuente. Todas las imágenes que fueron utilizadas se cargaron mediante una URL, resulta muy cómoda y eficiente esta alternativa.

La clase JButton descende de AbstractButton define un pequeño API público adicional, la implementación del manejo de eventos depende del tipo de botón escogido y de cómo se utiliza. Generalmente, se implementa un action listener, que es notificado cada vez que el usuario pulsa el botón, en esta aplicación se siguió la misma política para el manejo de eventos con los botones.

El botón y la etiqueta son componentes atómicos, su papel no es contener otros componentes Swing, sino que actúan como entidades auto-suficientes que representan bits de información par el usuario. Frecuentemente, los componentes atómicos también obtienen entrada del usuario, tal es el caso de los como boxes (JComboBox), campos de texto (JTextField), y tablas (JTable).



**Figura 3: Uso de botones en SEPEAGPE.**

### 3.1.4 Otros contenedores

JPanel es sólo una de las varias clases de contenedores que se puedan utilizar. Existen otros contenedores de propósito especial.

Los paneles o JPanel fueron de gran utilidad, pues juegan un papel más visible e interactivo en el GUI de un programa, siendo un eslabón importante en el diseño del software.

Los jTextField fueron utilizados para que el especialista introduzca el número del ejercicio que el estudiante desee buscar.

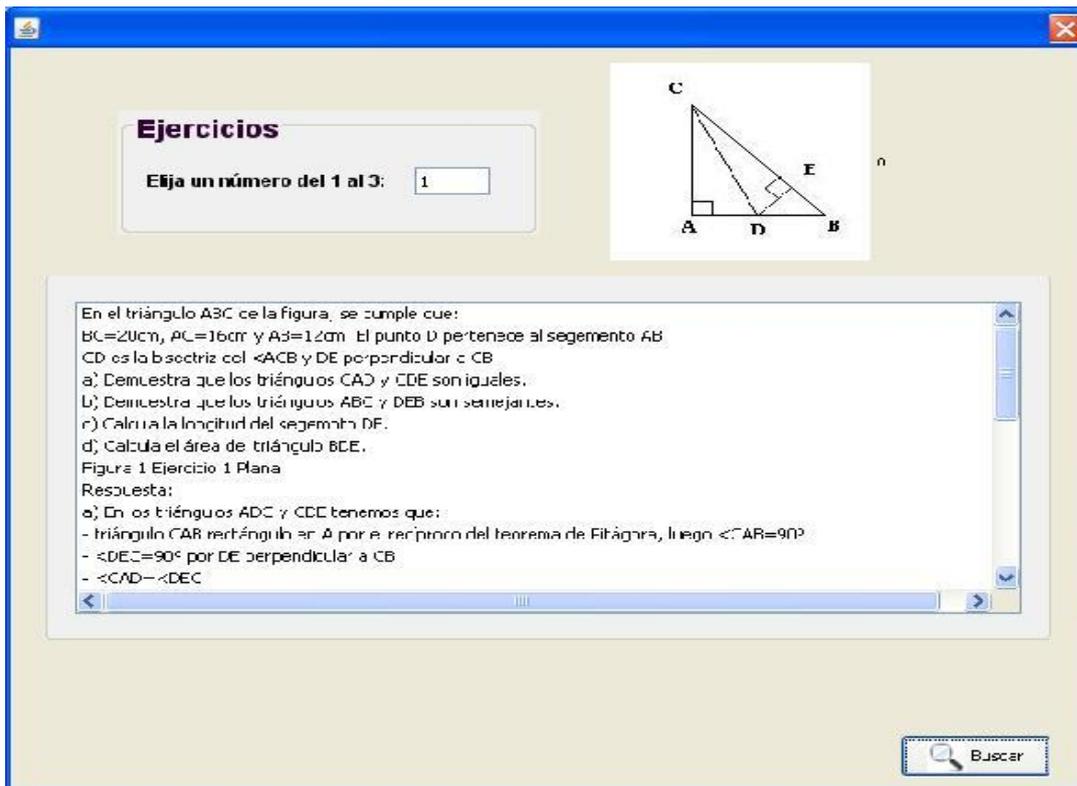


Figura 4: Uso de JTextField en SEPEAGPE.

### 3.1.5 Barra de menú para un JFrame

Los menús desplegables tienen detalles interesantes de implementación. Uno es que cada menú tienen un menú desplegable apropiado. Cuando el menú se activa, utiliza su menú desplegable para mostrar sus ítems de menú. Otro detalle es que un propio menú desplegable utiliza otro componente para implementar la ventana que contienen los ítems del menú.

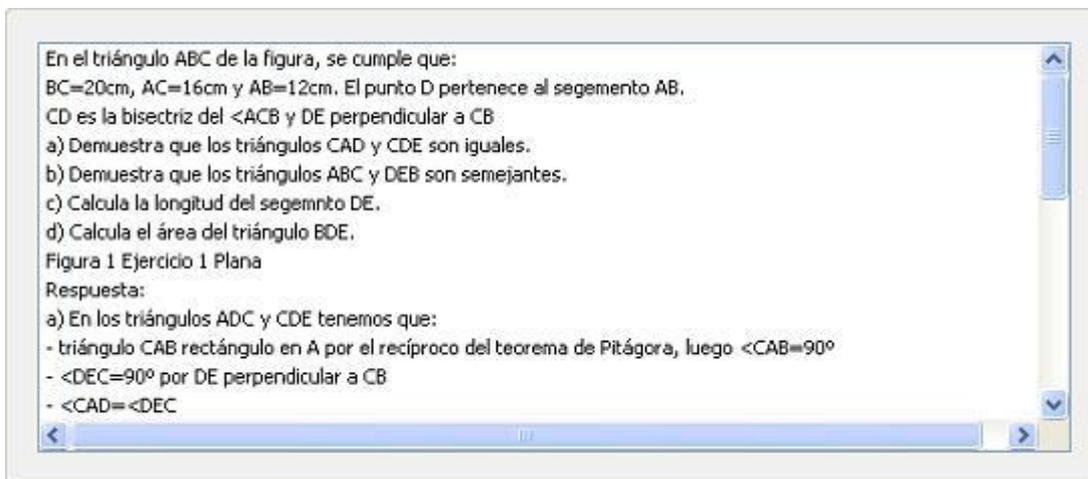
Para añadir un JMenu a un JMenuBar, se utiliza el método add(JMenu), mientras que para adicionar ítems de menú y submenús a un JMenu, se utiliza el método add(JMenuItem)



Figura 5: Barra de menú en SEPEAGPE.

### 3.1.6 JList

Los JList son utilizados para mostrar extensos textos extraídos de la base de conocimiento.



**Figura 6: Uso de JList en SEPEAGPE.**

### 3.2 Aspectos a tener en cuenta para la interacción de SWI-Prolog y Java

- Softwares necesarios:
  - SWI Prolog 5.6.33 o superior
  - Netbeans IDE 6.9.1 o superior
  - J2SE-JDK6 o superior
- Para hacer la interfaz se necesita una Base de Conocimiento, o sea un archivo fuente Prolog.

Aspectos fundamentales a seguir:

1. Crear un proyecto tipo Java con el Netbeans IDE.
2. Copiar la Base de Conocimiento (ejerciciop.pl) para el directorio raíz del proyecto.
3. Incorporar al proyecto la biblioteca jpl.jar ubicada en: C:\Archivos de programa\pl\lib\jpl.jar.

El éxito de la interfaz entre SWI Prolog y Java en gran medida depende de la configuración actual de su sistema operativo. Por tanto es preciso agregar a la variable

de entorno path seguida de los datos que trae por defecto un punto y coma y la siguiente línea:

- C:\Archivos de programa\Intel\DMX;C:\Archivos de programa\Java\jdk1.6.0\_20\bin; C:\Archivos de programa\Java\jdk1.6.0\_20\lib\tools.jar;C:\Archivos de programa\Java\jdk1.6.0\_20\jre\lib\rt.jar;C:\Archivos de programa\pl\bin; C:\Archivos de programa\pl\lib\jpl.jar;

En principio las condiciones están creadas para comenzar a inferir sobre el conocimiento, sin embargo es recomendable acatar las siguientes sugerencias:

1. Cerrar todas las aplicaciones, especialmente el Netbeans si lo tuviera abierto.
2. Limpiar el registro con alguna herramienta aparente como el CCleaner o TuneUp Utilities.
3. Reiniciar la máquina.

De esta manera queda enlazado el archivo fuente ejerciciop.pl con el proyecto Java. Para interrogar a la Base de Conocimiento se utiliza una instancia de la clase Query

1. Query query = new Query(String pathFile): Se llama una sola vez al iniciar la aplicación para cargar en memoria la Base de Conocimiento. Query query = new Query(String consult): Se llama cada vez que desea hacer una nueva consulta Prolog.

La clase Query tiene muchas funciones o métodos que permiten extraer inferencias del conocimiento representado en la Base de Conocimiento Prolog. Sin embargo con solo cinco funciones es posible implementar la interfaz de Prolog con Java de una manera flexible y cómoda. Estas funciones se describen a continuación:

- ✓ boolean hasSolution(): Método que permite saber si una consulta Prolog puede ser resuelta con el conocimiento disponible.
- ✓ Hashtable oneSolutions(): Este método devuelve un objeto Hashtable con la información referente a una solución de la consulta Prolog.
- ✓ boolean hasMoreSolutions(): Método que permite saber si la consulta Prolog tiene más soluciones.

- ✓ Hashtable nextSolutions(): Si la consulta Prolog tiene más de una solución, este método permitirá encontrar la siguiente solución.
- ✓ Hashtable [] allSolutions(): Retoma un arreglo de objetos tipo Hashtable con la información referente a todas las soluciones existentes.

Estas funciones utilizan internamente predicados de segundo orden, por lo que es conveniente interrogar al conocimiento Prolog utilizando sentencias simples con el objetivo de alcanzar una implementación lo más óptima posible.

### **3.3 Requerimientos del sistema.**

Para el correcto funcionamiento del software se necesitan un mínimo de requerimientos técnicos tanto de hardware como de software, siendo un software desarrollado en Java que ofrece la ventaja de ser multiplataforma, pero el instalador está hecho para ser utilizado en Windows.

#### **Requerimientos de hardware:**

- Al menos 256 MB de memoria RAM
- La instalación básica necesita de 132 MB de espacio disponible en disco más de 150 MB para la instalación del Java RunTime Environment (JRE) Versión 6 si no está instalado, aunque pudiera requerir más dependiendo del tamaño que pueda tener la base de conocimiento a medida que se vaya actualizando.
- Computador Pentium de 512MHz o superior.

#### **Requerimientos de software**

- Sistema Operativo Windows 2000, XP, 2003, Vista (x86, x64), Windows 7 (x86, x64).
- Máquina virtual de Java (JRE) en su versión 6 o mayor.
- SWI-Prolog versión 5.6.33 o superior.

#### **3.3.1 Facilidades que brinda al usuario.**

El software SEPEAGPE ofrece una interface gráfica de usuario GUI amigable, que constituye una herramienta que facilita una rápida utilización del SE. Se puede realizar inferencias de forma sencilla solo en varios segundos se puede obtener un resultado.

## Descripción de la aplicación.

SEPEAGPE posee una ventana principal. Esta es la ventana que se muestra al iniciar la aplicación, la cual muestra una barra menú donde se encuentran agrupadas las opciones que se le brinda al usuario.



Figura 7: Pantalla principal de SEPEAGPE.

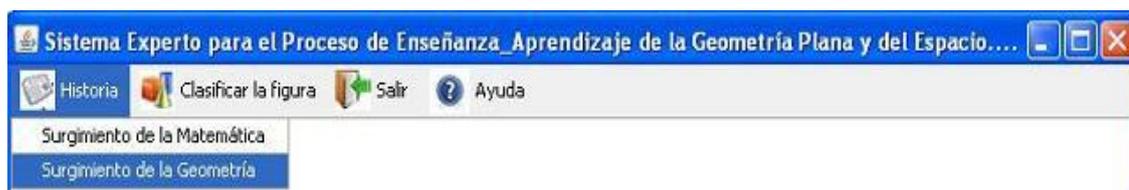
### 3.3.2 Evento action

El Evento action es el más utilizado en los SE. Los más sencillos y comunes manejadores de eventos. Se implementa un oyente de action para responder a una indicación del usuario de que alguna acción dependiente de la implementación debería ocurrir. Envía un mensaje **actionPerformed** a través del método **void actionPerformed (ActionEvent)** que se le llama justo después de que el usuario informe al componente escuchado de que debería ocurrir una acción.

### 3.3.3 Menús

Varias de las tareas que ofrece SEPEAGPE se inician a través de las opciones de la barra de menú. En la ventana se muestran las siguientes opciones de la barra de menú.

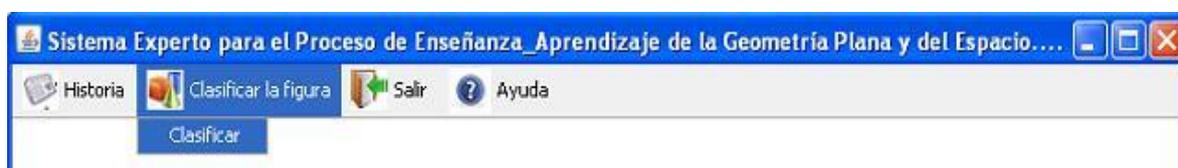
➤ **Menú Historia**



**Figura 8: Menú Historia**

Menú Historia el cual permite seleccionar, si desea ver el surgimiento de la Matemática o de la Geometría.

➤ **Menú Clasificar la figura**



**Figura 9: Menú Clasificar la figura**

Menú Clasificar la figura este menú al ser seleccionado permite clasificar la figura de acuerdo a las respuestas que usted le diga al sistema.

➤ **Menú Ayuda**



**Figura 10: Menú Ayuda**

Menú Ayuda, le permite al usuario acceder a la ayuda del software.

Para lograr una mayor eficiencia en el proceso de trabajo, y sobre todo para lograr una coherencia formal entre todos los módulos del sistema, se han pautado una serie de elementos comunes que facilitan su uso y reconocimiento. Se diseñó una pantalla inicial global del Sistema Experto, desde la cual se accederá a los diferentes módulos de este. El diseño está determinado fundamentalmente por el principio de usabilidad, cuyo propósito es facilitar su uso y comprensión manteniendo pautas estéticas y

agradables. El diseño de la interfaz es muy simple para el usuario y con buena resolución.

### **Principios de diseño**

El diseño de las interfaces de las aplicaciones está altamente condicionado por el medio para el cual se confeccionen. Los estándares existentes para los sistemas stand alone, difieren grandemente de las aplicaciones en ambiente Web, estos influyen notablemente en el éxito o fracaso de una aplicación. A continuación se describen los principios de diseño seguidos para el desarrollo de SEPEAGPE.

#### **3.3.4 Estándares en la interfaz de la aplicación.**

Para esta aplicación se desarrolló el diseño de la interfaz, basándose en el estándar de ventanas de Windows, de forma amigable y sencilla para posibilitar a los usuarios un ambiente conocido y por tanto fácil a la hora de utilizarlo. De modo general se utilizaron colores agradables a la vista, siguiéndose el mismo prototipo para todas las ventanas.

#### **3.3.5 Concepción general de la ayuda.**

El sistema cuenta con una ayuda capaz de explicar al usuario cómo es la manipulación y funcionamiento del software. La ayuda está formada por un menú interno que contiene todas las funcionalidades del sistema. El usuario podrá navegar por cada uno de ellos y consultarlo en cualquier momento ya que se encuentra ubicada en el menú principal de la aplicación.

#### **3.3.6 Tratamiento de excepciones.**

Una aplicación debe ser diseñada para evitar que se produzcan errores graves y que ante un error sea capaz de detectarlo y tratar de corregirlo. El correcto funcionamiento de todos los programas está dado en gran medida por la adecuada manipulación que se haga de los datos, teniendo en cuenta cada uno de los posibles errores que pueden ser introducidos en el sistema por los usuarios. Se dice que una aplicación bien diseñada minimiza el número y la peligrosidad de los errores, el control de estos, las validaciones necesarias y el tratamiento de excepciones, constituyen premisas que garantizan un buen desarrollo del software y la integridad de la información presentada.

Siempre que sea posible se debe recurrir a los controles de selección; como son: los botones de opción (radio buttons y combo box), el radio buttons es muy utilizado en SEPEAGPE. Por ejemplo para el trabajo con los botones de opción siempre habrá que seleccionar uno. De manera que si el usuario no selecciona ninguno y da aceptar da margen al error.

En cada uno de estos se le muestra un mensaje lo más claramente posible al usuario, informándole acerca de lo que acaba de ocurrir, o sea porque ocurrió el error.

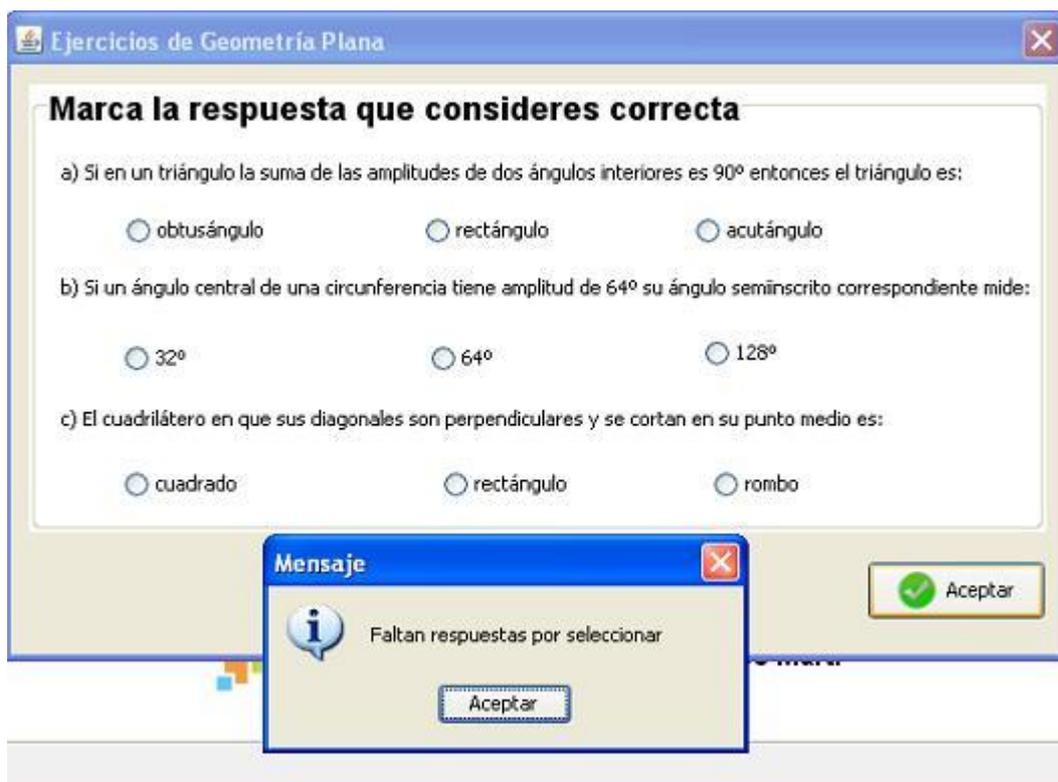


Figura 11: Uso de excepciones en SEPEAGPE.

### 3.4 Conclusiones

Se explicaron los principales requerimientos del sistema detallando como debe agregarse a la variable de entorno path las direcciones requeridas. Se determinó para la unión de Java con Prolog el uso de la clase Query de la librería jpl, siendo la componente que permite esta unión a través de las funciones que tiene implementada. Se eligieron los principales componentes que para el diseño de SEPEAGPE acordando la utilización de los JFrame, los JDialog, los RadioButtons y los JList, además JMenuItem y JMenu agregados a la barra JMenuItem. Se concertó el trabajo con el tratamiento de errores evitando un mal funcionamiento del software y se diseñó una ayuda para mostrar al usuario como trabajar con el sistema.

## Conclusiones

- 1) El estudio de los fundamentos teóricos y metodológicos evidenció que, la metodología utilizada y el lenguaje de programación Prolog, resulta apropiado para la construcción de sistemas expertos.
- 2) Se seleccionó la metodología del Dr. Mateo Lezcano (Lezcano Brito, 1995) para la construcción del sistema experto y de forma complementaria el Lenguaje Unificado del Modelado.
- 3) Se implementó un sistema experto para el Proceso de Enseñanza - Aprendizaje de la Geometría Plana y del Espacio, utilizando como herramienta para desarrollar la base de conocimiento SWI-Prolog y para la interfaz visual NetBeans 6.9.1 .

## **Recomendaciones**

Incorporar el contenido completo de la Geometría Plana y del Espacio para que los estudiantes de 12 grado puedan utilizar el sistema experto

## Bibliografía

- Definición de matemática.* (2013). Recuperado el 8 de enero de 2014, de Definición de matemática.
- Ana Lilia Laureano Cruces, F. d. (31 de marzo de 2001). *Técnicas de Diseño en Sistemas de Enseñanza Inteligentes.* Recuperado el 21 de abril de 2014, de Técnicas de Diseño en Sistemas de Enseñanza Inteligentes.
- autores, C. d. (2013). *Sistema Experto en el Proceso de Enseñanza-Aprendizaje de las ciencias exactas.* Durango, México.
- autores, C. d. (s.f.). *Enseñanza de las Matemáticas por medio de Sistemas Tutoriales.*
- autores, C. d. (s.f.). Sistema Experto para la recomendación de modelos instruccionales: una propuesta para su desarrollo. *Revista de Tecnología de Información y Comunicación en Educación*, 15.
- Brito, M. L. (1995). *Prolog y los Sistemas Expertos.* Santa Clara: Universidad Central de Las Villas.
- Brito, v. P. (1998). *Sistema para la Enseñanza de Sistemas Expertos.* Recuperado el 5 de marzo de 2014, de Sistema para la Enseñanza de Sistemas Expertos.
- Cando, Y. (19 de noviembre de 2012). *Metodología Orientada a Objetos.* Recuperado el 8 de enero de 2014, de Metodología Orientada a Objetos.
- Castellano, D. (2002). *Aprender y enseñar en la escuela: una concepción desarrolladora.* Recuperado el 8 de enero de 2014, de Aprender y enseñar en la escuela: una concepción desarrolladora.
- Castillo, E. (1999). *Sistemas Expertos y los Modelos de Redes Probabilísticas.* Recuperado el 8 de enero de 2014, de Sistemas Expertos y los Modelos de Redes Probabilísticas.
- Dinarte, E. (2 de febrero de 2011). *Las ramas de las matemáticas.* Recuperado el 8 de enero de 2014, de Las ramas de las matemáticas.
- Inteligencia Artificial.* (s.f.). Recuperado el 7 de noviembre de 2013, de Inteligencia Artificial.
- K., E. R. (1994). *Inteligencia Artificial.* Recuperado el 5 de marzo de 2014, de Inteligencia Artificial.
- Martin, M. (2010). *Sistema Experto para la clasificación temprana de Dengue Severo .* Obtenido de Sistema Experto para la clasificación temprana de Dengue Severo .
- Matemáticas.* (s.f.). Recuperado el 7 de noviembre de 2013, de Matemáticas.
- Sandoval. (2006). *Proceso de Enseñanza-Aprendizaje.* Recuperado el 5 de marzo de 2014, de Proceso de Enseñanza-Aprendizaje.

*Sistemas Expertos*. (s.f.). Recuperado el 7 de noviembre de 2013, de Sistemas Expertos.

System, P. S. (2005). *Modelado de sistemas con UML*. Recuperado el 8 de enero de 2014, de Modelado de sistemas con UML.

Tito, A. &. (2006). *Inteligencia Artificial*. Recuperado el 8 de enero de 2014, de Inteligencia Artificial.

Valverde, N. K. (s.f.). *Tutoriales Inteligentes aplicando Sistemas Expertos*.