

**SOFTWARE DE APOYO AL APRENDIZAJE EN LÍNEA
DE FORMAS BÁSICAS DE REPRESENTACIÓN DEL CONOCIMIENTO EN
INTELIGENCIA ARTIFICIAL
SARAI**

**ANDREA DUARTE DAZA
ELISA LILIANA MORALES QUINTERO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2005**

**SOFTWARE DE APOYO AL APRENDIZAJE EN LÍNEA
DE FORMAS BÁSICAS DE REPRESENTACIÓN DEL CONOCIMIENTO EN
INTELIGENCIA ARTIFICIAL
SARAI**

**ANDREA DUARTE DAZA
ELISA LILIANA MORALES QUINTERO**

**Trabajo de Grado presentado como requisito parcial para optar al título de
Ingeniería de Sistema**

**Directora
MARTHA VITALIA CORREDOR MONTAGUT
Dra. Ingeniería del Conocimiento**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2005**

AGRADECIMIENTOS

Las autoras expresan sus agradecimientos:

A la Doctora Martha Vitalia Corredor Montagut, por su valioso apoyo y colaboración en el proceso de elaboración de este proyecto, a pesar de sus múltiples ocupaciones y por ser una excelente profesional y persona digna de imitar.

A la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander y a todos sus docentes quienes nos aportaron sus conocimientos y valores.

A la Universidad Industrial de Santander por proporcionarnos la formación integral como personas y profesionales a lo largo de toda la carrera.

A nuestros familiares y amigos que siempre nos acompañaron y compartieron momentos especiales.

A todas las personas que de alguna u otra forma colaboraron con la realización de este proyecto en especial a Victor Hugo.

DEDICATORIA

A Dios, por ser el motor y guía de mi vida.

A mis Padres y Hermanos, Mercedes, Gonzalo, Christian, Gonzalo Alfonso y Yenny, por su amor incondicional, su gran apoyo en todos los momentos de mi vida y por sus valiosas enseñanzas.

A mi Familia, por su constante interés en ayudarme a alcanzar esta meta.

A Juan Carlos Gualdrón por su compañía y gran colaboración.

GRACIAS
Elisa Liliana

Dedicatoria

A DIOS porque a EL se lo debo TODO,

A mis padres Luis Alirio y Yolanda, por toda una vida de valiosas enseñanzas, mis hermanos, Milena, Gerson y Deicy, porque sé que siempre podré contar con ellos.

A Carlos Saúl, por todos estos años de apoyo, confianza y amor, te quiero mucho.

A Santiago, porque su llegada llenó mi vida de felicidad.

A mis gallinitas porque siempre estuvieron ahí... y

A Lilitiana, por tanta paciencia.

ANDREA

RESUMEN

TÍTULO

SOFTWARE DE APOYO AL APRENDIZAJE EN LÍNEA DE FORMAS BÁSICAS DE REPRESENTACIÓN DEL CONOCIMIENTO EN INTELIGENCIA ARTIFICIAL, SARAI*

AUTORES

ANDREA DUARTE DAZA

ELISA LILIANA MORALES QUINTERO**

PALABRAS CLAVES

Software

Inteligencia Artificial

Representación del Conocimiento

Educación Virtual

DESCRIPCIÓN

Este proyecto planteó como objetivo general el desarrollo de una herramienta software para reforzar el proceso de aprendizaje del estudiante de Ingeniería de Sistemas en el tema de Representación del Conocimiento en Inteligencia Artificial, que apoyará la orientación suministrada por el docente, permitiendo de esta manera la interacción estudiante y profesor. Ofrece el servicio de una base de datos, utilizada para subir y descargar documentos, además de toda la documentación sobre Redes Semánticas, Frames, Lógica de Predicados y Reglas de Producción, ejercicios resueltos y propuestos para que el usuario pueda aplicar los conceptos sobre la temática. Esta herramienta valida dos tipos de usuario profesor y estudiante, a quienes da la posibilidad de ingresar, visualizar o eliminar documentos (profesor) o solamente visualizarlos (estudiante); Las demás opciones ofrecidas son similares para los dos tipos de usuario.

Para su desarrollo se utilizó la metodología de prototipado evolutivo donde el diseño se realizó en UML ya que ayudó a comprender y expresar de mejor manera las necesidades del cliente; para su implementación se utilizó lenguaje de libre distribución que constituye una buena alternativa para el desarrollo de software de bajo costo y finalmente se implantó en la Web en la plataforma del Aula Virtual con el fin de poder llegar a todos los usuarios que tengan previos conocimientos acerca del tema.

Al implantar la herramienta se ampliaron las capacidades que posee el Aula Virtual y dio pautas para el inicio de la creación de clases virtuales en la Universidad Industrial de Santander, con el propósito de ir a la par en la constante evolución tecnológica donde las necesidades del ser humano varían con el tiempo.

* Trabajo de Grado.

** Facultad de Ingenierías FísicoMecánicas, Ingeniería de Sistemas, Dir. Dra Martha Vitalia Corredor Montagut.

SUMMARY

TITLE

Support software to the online learning of basic forms of knowledge representation artificial intelligence, SARAI*

AUTHORS

ANDREA DUARTE DAZA

ELISA LILIANA MORALES QUINTERO**

KEYWORDS

Software

Artificial intelligence

Knowledge representation

Virtual education

ABSTRACT

This degree project had as a main objective the development of tool software for reinforcing the learning process for the systems engineering student in the subject of knowledge representation in artificial intelligence, which will support the guide given by the teacher. This will improve the interaction between teacher and student. The tool software offers a database used to upload and download documents, and all the documentation about semantic networks, frames, logic of predicates and production rules solved and proposed problems for the user to apply concepts about the subject. This tool validate two kind of users, teacher and student, which have the possibility to log in, visualize, or delete documents (teacher) or just visualize them (student); the rest of the options offered are similar for both kind of users.

For its development was used a evolutive prototyped methodology where the design was made in UML because it helped to understand and express in a better way the client needs; for its implementation was used a freeware language which constitutes a good choice for the development of low cost software and finally it was established in the Web in the platform of the Virtual Classroom with the main idea to reach all the users with previous knowledge about the subject.

When establishing the tool the capacities of the virtual classroom were expanded and it gave guidelines for the beginning of the creation of virtual classes in the Universidad Industrial de Santander, with the purpose to be in the latest technological evolutions where the human being needs changes every time.

* Degree Project.

** FísicoMecánicas engineering faculty, Systems engineering, Dir. Dra. Martha Vitalia Corredor Montagut.

INTRODUCCIÓN

En los últimos años los avances tecnológicos y su acelerado crecimiento han motivado a que las diferentes áreas de desempeño, se enfoquen en la utilización de nuevas estrategias que utilicen las tecnologías de la computación con el fin de aprovechar todos los recursos que estas ofrecen, respondiendo así a las necesidades cambiantes del hombre y poder seguir a la par de un mundo donde la computación y la tecnología son parte del quehacer humano. Son estas necesidades las que han obligado a las instituciones educativas a replantear sus estrategias de enseñanza y ofrecer experiencias que usen los servicios de las Tecnologías de la Información y Comunicación (TIC). El propósito al introducir las TIC como apoyo en procesos educativos es enriquecer las experiencias de aprendizaje con el uso de sus servicios para apoyar en el estudiante el acceso y la organización adecuada de información, el intercambio de gran cantidad de información, el refuerzo de las funciones cognitivas, el ejercicio de las operaciones mentales y el protagonismo de los estudiantes, fomentando en ellos la responsabilidad de su propio proceso de aprendizaje a partir de la información y los conocimientos que le son presentados de una manera dinámica y que le exigen una permanente actividad y participación. En este sentido, las TIC no hace que los docentes dejen de ser indispensables, sino que modifica su papel en relación con el proceso de aprendizaje.

En la dirección de ofrecer alternativas adicionales de aprendizaje a los estudiantes de Ingeniería de Sistemas que cursan la asignatura de Inteligencia Artificial, se creó la herramienta objeto de este proyecto, que pretende apoyar, enriquecer y fortalecer el proceso de aprendizaje del estudiante sobre las diferentes formas de representación del conocimiento, utilizadas en los desarrollo y aplicaciones de la Inteligencia Artificial.

TABLA DE CONTENIDO

INTRODUCCIÓN	1
1. PRESENTACIÓN DEL PROYECTO	8
1.1 PLANTEAMIENTO DEL PROBLEMA	8
1.2 OBJETIVOS DEL PROYECTO	10
1.2.1 Objetivo General	10
1.2.2 Objetivos Específicos.....	10
1.3 JUSTIFICACION, IMPACTO Y VIABILIDAD	11
2 MARCO TEORICO	13
2.1 INTELIGENCIA ARTIFICIAL	13
2.2 REPRESENTACION DEL CONOCIMIENTO	14
2.2.1 Formas de Representación del Conocimiento	16
2.3 LOGICA DE PREDICADOS	16
2.3.1 Definición	16
2.3.2 Sintaxis	17
2.3.3 Reglas de Inferencia	21
2.3.4 Ventajas del uso de la lógica de Predicados.....	31
2.3.5 Desventajas del uso de la Lógica de Predicados.....	31
2.4 REGLAS DE PRODUCCIÓN	32
2.4.1 Definición	32
2.4.2 Reglas de Inferencia	33
2.4.3 Estrategias de Búsqueda	34
2.4.4 Ventajas de su Uso	41
2.4.5 Desventajas de su Uso	41
2.5 REDES SEMÁNTICAS	41
2.5.1 Definición	41
2.5.2 Componentes.....	42
2.5.3 Reglas de Búsqueda.....	43
2.5.4 Ventajas de las Redes Semánticas.....	46
2.5.5 Desventajas de las Redes Semánticas.....	46
2.6 FRAMES	47
2.6.1 Definición	47
2.6.2 Componentes.....	47
2.6.3 Reglas de Búsqueda.....	49
2.6.4 Ventajas de los Frames	51
2.6.5 Desventajas de los Frames.....	52

2.7 EDUCACIÓN E INFORMATICA	52
2.7.1 EDUCACIÓN VIRTUAL	53
2.8 INGENIERIA DEL SOFTWARE	55
2.8.1 Prototipado Evolutivo	56
2.8.2 Lenguaje de Modelado Unificado (UML).....	57
2.9 TECNOLOGÍAS WEB	61
2.9.1 HTML	61
2.9.2 PHP.....	62
2.9.3 JAVASCRIPT	64
2.9.4 JAVA.....	65
2.10 BASES DE DATOS	67
2.10.1 MYSQL	69
3. DISEÑO Y DESARROLLO DEL SISTEMA	70
FASE INICIAL	70
3.1 CONCEPTO INICIAL	70
3.1.1 Recopilación de Información.....	70
3.1.2 Análisis y Especificación de Requerimientos	70
3.1.3 Entrenamiento previo	72
FASE ITERATIVA	72
3.2 DISEÑO DE LA HERRAMIENTA	72
3.2.1 Diagramas de Casos de Uso	72
3.2.2 Diseño de Interfaz.....	76
3.2.3 Diseño de la Base de Datos.....	78
3.3 IMPLEMENTACION DEL PRIMER PROTOTIPO	79
3.3.1 Código Fuente PHP para Conexión con la Base de Datos	79
3.3.2 Código Fuente PHP para Cerrar la Conexión con la Base de Datos	79
3.3.3 Código Fuente que Permite el Ingreso de Nuevos Documentos	80
3.3.4 Código Fuente que Permite Eliminar Documentos	80
3.3.5 Código Fuente que Permite Visualizar Documentos.....	81
3.3.6 Código Fuente para Validar el Tipo de Usuario	82
3.4 PRUEBAS DEL PRIMER PROTOTIPO	82
3.4.1 Entrega del primer prototipo.....	82
3.5 DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO FINAL	83
FASE FINAL	84
3.6 IMPLANTACIÓN DEL PROTOTIPO FINAL	84
3.6.1 Código Fuente para Acceder a la Herramienta.....	84
3.6.2 Requisitos para el uso de la herramienta.....	86
4. DESCRIPCIÓN DE LA HERRAMIENTA	87

4.1 PANTALLA DE INICIO.....	87
4.2 PANTALLA PRINCIPAL	88
4.2.1 Frame Superior	88
4.2.2 Frame Central	89
4.2.3 Frame Lateral.....	89
4.3 PANTALLAS DE LAS FORMAS DE REPRESENTACIÓN DEL CONOCIMIENTO.....	90
4.3.1 Pantalla Redes Semánticas	90
4.3.2 Pantalla Reglas de Producción	95
4.3.3 Pantalla Frames.....	98
4.3.4 Pantalla Lógica de Predicados.....	99
5. CONCLUSIONES	103
6. RECOMENDACIONES.....	105
BIBLIOGRAFÍA.....	106

LISTA DE FIGURAS

Figura 1. Proceso de razonamiento progresivo.....	36
Figura 2. Ciclaje en Búsqueda hacia atrás	38
Figura 3. Extensión	39
Figura 4. Profundidad	40
Figura 5. Concepto de Red Semántica	42
Figura 6. Ejemplo de Red Semántica	44
Figura 7. Operación de Confrontación en redes	45
Figura 8. Esquema de un Frame	48
Figura 9. Prototipado Evolutivo.....	57
Figura 10. Diagrama de Casos de Uso generales del Aula Virtual	73
Figura 11. Actores que intervienen en SARAI	74
Figura 12. Diagrama de Casos de Uso – Estudiante	74
Figura 13. Diagrama de Casos de Uso - Profesor	75
Figura 14. Diagrama Casos de Uso General SARAI	75
Figura 15. Interfaz Gráfica Inicial	76
Figura 16. Interfaz Gráfica General.....	77
Figura 17. Tercer Interfaz Gráfica – Profesor	77
Figura 18. Tercer Interfaz Gráfica – Estudiante	78
Figura 19. Vínculo a la ventana Herramientas	84
Figura 20. Ubicación de SARAI en el Aula Virtual	85
Figura 21. Pantalla de Inicio	87
Figura 22. Pantalla Principal	88
Figura 23. Frame Superior de Pantalla Principal	88
Figura 24. Frame Central de Pantalla Principal	89
Figura 25. Frame Lateral de Pantalla Principal.....	89
Figura 26. Pantalla Redes Semánticas.....	90
Figura 27. Pantalla Ejercicios Redes Semánticas Estudiante.....	91
Figura 28. Pantalla Ejercicios Redes Semánticas Profesor	92
Figura 29. Pantalla Ejercicio a Resolver Redes Semánticas	93
Figura 30. Applet de Ejercicio a Resolver de Redes Semánticas	94
Figura 31. Ventana de Respuesta Errada.....	95
Figura 32. Pantalla Reglas de Producción.....	95
Figura 33. Pantalla Ejercicio a Reglas de Producción	96
Figura 34. Primer Ejercicio a Resolver Reglas de Producción.....	97
Figura 35. Segundo Ejercicio a Resolver Reglas de Producción	97
Figura 36. Pantalla Frames.....	98
Figura 37. Pantalla Enunciado Ejercicio Frames	98
Figura 38. Ejercicio a Resolver Frames	99
Figura 39. Pantalla Lógica de Predicados	99
Figura 40. Pantalla Ejercicio a Resolver Lógica de Predicados	100
Figura 41. Primer Ejercicio Lógica	100
Figura 42. Segundo Ejercicio Lógica	101

Figura 43. Tercer Ejercicio Lógica Pantalla inicial..... 102
Figura 44. Tercer Ejercicio Lógica 102

LISTA DE TABLAS

Tabla 1. Conectores Lógicos	20
Tabla 2. Campos de las tablas de la Base de Datos	79

1. PRESENTACIÓN DEL PROYECTO

1.1 PLANTEAMIENTO DEL PROBLEMA

Los constantes cambios tecnológicos y su acelerada expansión a todos los campos, han exigido que la educación se oriente hacia nuevas formas de pedagogía con el fin de aprovechar todos los recursos que nos ofrecen las tecnologías, así como responder a las necesidades actuales de los estudiantes. Y es precisamente estos cambios los que han obligado a las instituciones educativas a ofrecer experiencias que usen los servicios de las tecnologías de la información para apoyar las clases presenciales, mediante la presentación de contenidos, en la cual el estudiante disponga de la información en cualquier momento, mantenga una constante interacción con sus compañeros y profesor y logre aprendizajes significativos¹, experiencias que en todo momento buscan la calidad de la educación.

Actualmente, algunas asignaturas, dentro de las cuales está Inteligencia Artificial², ofrecen a los estudiantes matriculados, un ambiente en Internet llamado AULA VIRTUAL³, el cual brinda servicios de foros, cronograma de actividades, contenidos, talleres, fechas de entregas de trabajos, publicaciones de trabajos de clase destacados, presentación de la temática de la materia, correo electrónico, Chat, entre otros, con el fin de poder mantener un contacto y una dinámica de trabajo entre los estudiantes y el docente. Sin embargo, con respecto a Inteligencia Artificial, en este momento no se cuenta con una alternativa que

¹ Aprendizaje Significativo, según Ausubel, es aquel en donde el alumno relaciona lo que ya sabe con los nuevos conocimientos, es decir sus experiencias representan un factor de mucha importancia.

² Inteligencia Artificial, es una Electiva Técnica Profesional, ofrecida por la Escuela de Ingeniería de Sistemas e Informática, de la Universidad Industrial de Santander.

³ Aula Virtual: <http://tic.uis.edu.co/aula>

refuerce los contenidos de la asignatura, en especial el tema de Representación del Conocimiento, observando como principal inconveniente para los estudiantes, la sintaxis y semántica de sus formas básicas, como son las Reglas de Producción, las Redes Semánticas, la Lógica de Predicados y los Frames. Es por esto que se ve la necesidad de crear una herramienta que pueda apoyar el proceso de aprendizaje del estudiante y además favorezca:

- ❖ La profundización del tema dada la limitación de horas de clase.
- ❖ La interacción permanente entre profesor y estudiantes.
- ❖ El acceso a información proveniente de diversas fuentes

Los posibles criterios de solución a considerar en el manejo de estos aspectos son:

- ❖ Una solución que permita unificar conceptos de los diferentes temas tratados en el desarrollo de la clase.
- ❖ Una solución disponible en el momento y lugar que el estudiante lo requiera.
- ❖ Una solución que ayude al estudiante a reforzar sus conocimientos por medio de una fundamentación más explícita y lo ponga en contacto con diversas fuentes de información.

1.2 OBJETIVOS DEL PROYECTO

1.2.1 Objetivo General

Desarrollar una herramienta software, que facilite la comprensión de formas de Representación del Conocimiento, como apoyo a la adquisición de conceptos básicos para los desarrollos en Inteligencia Artificial.

1.2.2 Objetivos Específicos

- ❖ Identificar y describir las principales características en cuanto a las reglas de sintaxis y de formación, así como los mecanismos de inferencia utilizados en Redes Semánticas, Lógica de Predicados, Reglas de Producción y Frames, como formas básicas de Representación del Conocimiento, utilizados en los desarrollos de Inteligencia Artificial.

- ❖ Diseñar, implementar e implantar **SARAI** como herramienta específica de la plataforma AULA VIRTUAL de la Universidad Industrial de Santander, teniendo en cuenta los lineamientos de software y arquitectura que en ella se utiliza y que permita:
 - a. Presentar la fundamentación teórica de la Representación del Conocimiento y sus formas básicas de Reglas de Producción, Lógica de Predicados, Redes Semánticas y Frames.

 - b. La actualización de contenidos, ejercicios y problemas sobre las formas básicas de Representación del conocimiento utilizados en Inteligencia Artificial.

- c. Acceder a ejemplos y aplicaciones que desarrollen en el estudiante, habilidades en la solución de problemas relacionados con la Representación del Conocimiento en el área de Inteligencia Artificial.
- d. Proponer ejercicios a los estudiantes y ofrecer la realimentación correspondiente una vez resueltos éstos.

1.3 JUSTIFICACION, IMPACTO Y VIABILIDAD

La buena preparación de los alumnos en cada una de las asignaturas que cursan en la universidad, es de vital importancia para sus prácticas estudiantiles y más adelante para su desempeño profesional; el uso de nuevas tecnologías para apoyar el aprendizaje del alumno ha favorecido el interés de los estudiantes por aprender, pues estas herramientas facilitan el acceso a gran cantidad de información actualizada, lo que en muchas ocasiones les facilita la comprensión de conceptos ya que pueden realizar investigaciones, comparaciones y relaciones entre los enfoques de diversos autores e investigadores.

La herramienta software a desarrollar apoyará la orientación suministrada por el docente sobre el tema de Representación del Conocimiento, en el área de Inteligencia Artificial, pues este conocimiento es indispensable para la solución de problemas donde se requiera el desarrollo de las cuatro formas básicas: Reglas de Producción, Redes Semánticas, Lógica de Predicados y Frames, que son las formas clásicas de representación y, por tanto, las más utilizadas y aceptadas entre expertos y son las estudiadas en el curso. Esta herramienta se plantea para mejorar el proceso de aprendizaje del alumno en este tema y ofrecerle una alternativa de estudio que tenga en cuenta las ventajas de las nuevas tecnologías, lo que seguramente aportará el enriquecimiento de los procesos de aprendizaje.

Al analizar la viabilidad de este proyecto, se tuvo en cuenta lo siguiente:

En la parte técnica, "SARAI" requiere de un computador personal para poder ejecutarse, que debe contar con especificaciones mínimas e Internet Explorer 4.0 superior o Netscape 4.0 superior, las cuales son comunes en el mercado actual.

En el aspecto económico, este proyecto se considera viable debido a la alta demanda de computadores y el fácil acceso a Internet, que hacen que los costos de estos recursos se disminuyan, logrando así que la mayoría de las personas tengan acceso a ellos. Adicionalmente, el software de desarrollo utilizado es de libre distribución, por lo que no genera costos.

En el aspecto social y educativo, se pretende reforzar con la herramienta una cultura de clases virtuales en la Universidad Industrial de Santander, con el fin de ir a la par en la constante evolución tecnológica, permitiendo con esto, extender las capacidades que posee Aula Virtual para apoyar procesos de aprendizaje.

2 MARCO TEORICO

2.1 INTELIGENCIA ARTIFICIAL

La Inteligencia Artificial (IA), es una nueva generación de tecnología informática, caracterizada no sólo por su arquitectura (*hardware*), sino también por sus capacidades; puede decirse que la Inteligencia Artificial (IA) es una de las áreas más fascinantes y con más retos de las Ciencias de la Computación. Nació como estudio filosófico de la inteligencia humana, mezclada con la inquietud del hombre de imitar la naturaleza circundante (como ver, caminar, volar, nadar etc.), hasta inclusive querer imitarse a sí mismo. Sencillamente, la Inteligencia Artificial busca imitar la inteligencia humana creando sistemas que aprenden nuevos conceptos y tareas, sistemas que pueden razonar y derivar conclusiones útiles acerca del mundo que nos rodea, sistemas que pueden comprender un lenguaje natural o percibir y comprender una escena visual, y sistemas que realizan otro tipo de actividades que requieren de inteligencia humana.

Las diferentes funciones inteligentes que puede emular esta área (y tal vez en algunos casos superar) a las del ser humano son:

Aprendizaje:

- Captación automática de conocimientos.

Razonamiento:

- Sistemas basados en conocimientos.
- Bases de datos inteligentes.
- Prueba de teoremas
- juegos.

Percepción:

- Comprensión de lenguaje natural.
- Interpretación de escenas visuales (Visión por computadora).

Locomoción y Manipulación:

- Realizar procesos mecánicos y tareas manuales (Robótica).

Creación:

- Generación, verificación, depuración y optimización automática de programas.

2.2 REPRESENTACION DEL CONOCIMIENTO

En los seres humanos se estima que el conocimiento es almacenado como estructuras complejas de neuronas interconectadas; en las computadoras, el conocimiento también se almacena como estructuras simbólicas, pero en forma de estados eléctricos y magnéticos. Para representar el conocimiento, el ser humano, se vale de formas simbólicas como: imágenes, lenguaje hablado y lenguaje escrito, adicionalmente, ha desarrollado otros sistemas de representación del conocimiento: literal, numérico, estadístico, estocástico, lógico.

El conocimiento que posee un ser humano se adquiere a través del tiempo. Este proceso requiere una construcción permanente de conceptos, lo que exige el desarrollo de acciones de compilación o almacenamiento, organización e indexación del cúmulo de teorías y heurísticos que va construyendo a lo largo de su vida, de manera que sea fácil su elección y recuperación cuando se necesitan en la solución de problemas. Este conocimiento se va construyendo de dos maneras, informal y formal.

El conocimiento formal se va adquiriendo por medio de la educación formal, los libros, las conferencias, revistas, etc., medios que posibilitan la apropiación de leyes, teorías y principios básicos, necesarios para justificar científicamente la solución de problemas. Por otro lado, el conocimiento informal se adquiere por medio de la experiencia, ya sea personal o del entorno, que permite la construcción de reglas prácticas para resolver rápida y eficientemente problemas, a partir de las bases conceptuales adquiridas formalmente.

Para resolver problemas que utilizan conocimiento, propios de la Inteligencia Artificial, como por ejemplo lograr que una computadora pueda razonar y resolver dudas de los usuarios, se requiere de grandes cantidades de estructuras conceptuales que puedan entender lo suficiente para saber qué objetos y hechos se tienen que tener en cuenta y cuáles ignorar en la solución de problemas. Este conocimiento se debe ordenar de tal forma que puede ser utilizado por una máquina para ofrecer una solución óptima a determinados problemas.

La implementación de este tipo de aplicaciones requiere de un sistema de Representación del Conocimiento en el cual cada componente tiene una función determinada, que combinada con las demás facilitan el logro de un fin específico, en este caso *“expresar el conocimiento en forma manejable por una computadora”*⁴.

Un buen sistema de Representación del Conocimiento debe poseer las siguientes propiedades:

- **Suficiencia de la Representación:** Capacidad de representar todos los tipos de conocimiento necesarios en el dominio.

⁴ RUSELL, Stuart. Inteligencia artificial un enfoque moderno. Primera edición. México, 1996.

- **Suficiencia Deductiva:** Capacidad para manipular las estructuras de la representación con el fin de obtener nuevas estructuras que correspondan con un nuevo conocimiento deducido a partir del primero.
- **Eficiencia deductiva:** Capacidad de incorporar información adicional en las estructuras de conocimiento con el fin de que los mecanismos de inferencia puedan seguir las direcciones mas eficientes.
- **Eficiencia en la adquisición:** Capacidad de adquirir nueva información con facilidad. Idealmente, el programa debería controlar la adquisición de conocimiento por sí mismo.

2.2.1 Formas de Representación del Conocimiento

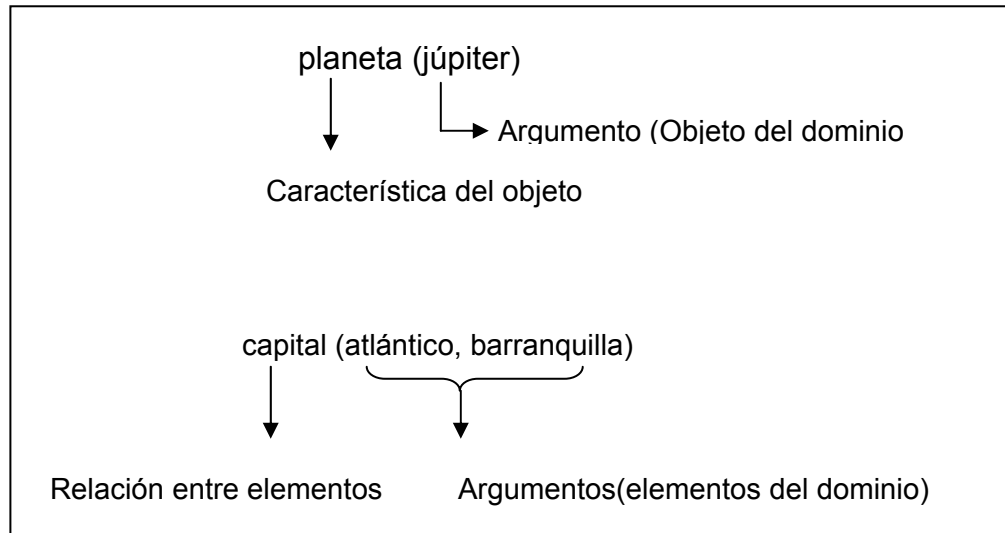
Debido a que todavía no se ha encontrado ningún sistema que optimice todos los aspectos presentados anteriormente y que sea aplicable a cualquier tipo de conocimiento, existen diversas técnicas o formas para la Representación del Conocimiento. Entre las formas clásicas de representación están la Lógica de Predicados, las Redes Semánticas, las Reglas de Producción y los Frames entre otras.

2.3 LOGICA DE PREDICADOS

2.3.1 Definición

La lógica de Predicados de primer orden es la lógica que tiene como elemento básico el predicado y es de primer orden porque solo cuantifica variables. El predicado es un enunciado que describe características y/o relaciones entre los objetos del mundo que se está trabajando.

Ejemplo:



Igualmente la Lógica de Predicados:

“Estudia la relación de implicación entre suposiciones y conclusiones”⁵. Y,

“Proporciona de manera inmediata un método muy potente para la obtención de nuevo conocimiento a partir del antiguo: la deducción matemática”⁶

2.3.2 Sintaxis

La sintaxis, abarca los elementos y las reglas de escritura de elementos a tener en cuenta para escribir sentencias correctas en lógica de predicados.

⁵ KOWALSKI, Robert, Lógica, Programación e Inteligencia Artificial. Madrid. 1986.

a. Alfabeto

La lógica de predicados ofrece un alfabeto que permite formalizar conocimiento de un dominio. Los elementos de este alfabeto que son usados en desarrollos computacionales son:

- **Constantes:** términos fijos que se usan para representar argumentos de predicados que no varían en un proceso de inferencia. Deberán escribirse con minúscula inicial.

Ejemplo: mineral (oro)

En este caso oro es una constante, las constantes pueden ser numéricas o simbólicas.

- **Variables:** términos usados para representar argumentos de predicados que pueden variar en un proceso de inferencia. Se escriben con letras mayúsculas y minúsculas, el “_” y dígitos del 0 al 9, iniciando con mayúscula.

Ejemplo: edad(carlos,X)

En este caso X es una variable que tomará como valor la edad de carlos.

- **Funciones:** términos que describen elementos identificándolos como el resultado único de la aplicación de una transformación entre otros elementos del dominio.

Ejemplo: menor(X,Y)

Evaluará las variables X y Y y dependiendo de sus valores devolverá si o no.

⁶ RICH Elaine, KNIGHT Kevin, Inteligencia Artificial, Madrid,1994 .

- **Cuantificadores:** encargados de generalizar o especificar la aplicación de predicados a ciertos elementos de un dominio.

- **Cuantificador universal;** (\forall) permite afirmar que una característica, o relación se cumple para todos los valores posibles de la variable que es cuantificada.

Ejemplo:

$(\forall X): \text{hombre}(X) \rightarrow \text{persona}(X),$

Significa que para todo X que sea hombre, implica que sea persona.

- **Cuantificador existencial;** (\exists) permite afirmar que por lo menos la relación o característica se cumple para algún valor de la variable asociada al operador.

Ejemplo:

$(\exists X): \text{primos}(X) \wedge \text{impares}(X),$

Significa que existe un X tal que sea primo e impar.

- **Conectores:** encargadas de formar expresiones compuestas, a partir de expresiones más simples. Las conectivas más utilizadas en representaciones de IA son:

- Unaria: Se antepone al predicado y se representa por medio de la negación (\sim); La negación implica que el predicado no se cumple para el o los argumentos.
- Binaria: son símbolos que se usan entre dos predicados. Los mas utilizados en Inteligencia Artificial

NOMBRE	CONECTOR	SÍMBOLO
Binarias:		
Conjunción	AND	\wedge
Disyunción	OR	\vee
Implicación	ENTONCES	\rightarrow
Equivalencia	SI, SOLO SI	\leftrightarrow

Tabla 1. Conectores Lógicos

- **Átomos:** predicados cuyos argumentos son constantes, variables o funciones. Constituyen la mínima expresión que puede escribirse en lógica.

b. Reglas de Formación

Las reglas de formación permiten la construcción de las llamadas fórmulas bien formadas o bien escritas en lógica (fbf), a partir de variables, constantes, conectores, cuantificadores, átomos, predicados y funciones.

Estas son:

- Todo átomo es una fbf en lógica.
- Uso de la negación: si “p” es una fbf en lógica, entonces $\sim p$ también lo es. Puede verse que se usa una notación prefija para el uso de conectores unarios, es decir, éstos se colocan antes de la fbf, que se va a negar.
- Uso de los conectores Implicación, equivalencia, disyunción y conjunción: si “p” y “q” son fbf en lógica y “D” es un conector cualquiera, entonces “p D q” es una fbf. En este caso, puede verse que con los conectores binarios, se utiliza una notación infija, es decir, se sitúa el conector entre las fbf que se van a enlazar.

- d. Uso de los cuantificadores: si “ $p(X)$ ” es una fbf en lógica, entonces, $\exists(X) p(X)$ o $\forall(X) p(X)$, también lo son. Por lo que se concluye que el uso de los cuantificadores sigue notación prefija, es decir, se colocan antes de la fbf donde estén los predicados cuyos argumentos cubren con su alcance

2.3.3 Reglas de Inferencia

Inferir es concluir o decidir a partir de algo conocido o asumido; llegar a una conclusión. A su vez, razonar es pensar coherente y lógicamente; establecer inferencias o conclusiones a partir de hechos conocidos o asumidos. El proceso de razonamiento, por lo tanto, involucra la realización de inferencias, a partir de hechos conocidos. Realizar inferencias significa derivar nuevos hechos a partir de un conjunto de hechos conocidos como verdaderos. La lógica de predicados proporciona un grupo de reglas sólidas, con las cuales se pueden realizar inferencias. Las principales **Reglas de Inferencia** son:

- a. **Modus ponens.**- Es la más importante, en los sistemas basados en conocimiento. Establece que:

Si las sentencias $(p(X) \rightarrow q(X))$ se conocen que son verdaderas,

Y se confirma la validez de $p(a)$

Entonces se puede inferir que $q(a)$ también es verdadera.

Ejemplo:

$(\forall X): \text{mamífero}(X) \rightarrow \text{vivíparo}(X)$ es verdadero

y también es verdadero que: *mamífero (ballena)*

Luego es verdadero vivíparo (ballena)

Este tipo de razonamiento exige mucha exactitud en la construcción de las reglas, además de una fuerte relación entre antecedentes y consecuentes.

b. Modus tolens.- Esta regla establece que:

*Si la sentencia $(p(X) \rightarrow q(X))$ es verdadera y
Se confirma la validez de $\sim q(a)$
Entonces se puede concluir $\sim(p(a))$*

Ejemplo:

*$(\forall X): \text{servivo}(X) \rightarrow \text{mortal}(X)$ es verdadero y
se establece que: $\sim\text{mortal}(\text{piedra})$ es verdadero,
Entonces, $\sim\text{servivo}(\text{piedra})$*

c. Resolución: *permite* obtener todas las posibles conclusiones a partir de un conjunto de premisas escritas según las reglas de la lógica, que constituyen el conocimiento sobre un dominio o una situación. La resolución solamente se aplica a frases en forma clausulada (predicados unidos por disyunciones), teniendo en cuenta que las frases escritas en lógica se expresan utilizando todos los símbolos del alfabeto de la lógica, es necesario realizar un procedimiento de conversión de estas frases a la forma clausulada.

Procedimiento para convertir fbf en cláusulas:

a. Eliminar las implicaciones y equivalencias:

$a \rightarrow b \equiv (\sim a \vee b)$ Leyes de la Implicación

$a \leftrightarrow b \equiv (\sim a \vee b) \wedge (\sim b \vee a)$ Leyes de la doble Implicación

b. Introducir las negaciones de manera que solamente afecten a los predicados

➤ **Leyes de Morgan**
 $\sim (a \wedge b) \equiv (\sim a \vee \sim b)$
 $\sim (a \vee b) \equiv (\sim a \wedge \sim b)$

➤ **Negación de Cuantificadores**
 $\sim (\exists X) p(X) \equiv (\forall X) \sim p(X)$
 $\sim (\forall X) q(X) \equiv (\exists X) \sim q(X)$

➤ **Negación de Negación**
 $\sim (\sim a) \equiv a$

c. Normalizar las variables de forma que cada cuantificador esté ligado a una única variable.

$$(\forall X) p(X) \vee (\forall X) q(X)$$

se convierte en

$$(\forall X) p(X) \vee (\forall Y) q(Y)$$

d. Eliminar los cuantificadores existenciales, en este paso pueden presentarse dos situaciones.

- Si el \exists no está dentro del dominio de un \forall .

Ejemplo: Representar la afirmación “alguna mujer es madre de Luisa” se tendría

$$(\exists X) \text{ mujer}(X) \wedge \text{madre}(X, \text{luisa})$$

La eliminación del \exists , según el procedimiento, consiste en idearse lo que se llama “**constante de Skolem**”. Este paso es el que menos justificación matemática

tiene; se sustenta en que como existe algún elemento del dominio que cumple con la propiedad y/o relación que se refiere a la variable del existencial, se inventa su nombre y se sigue adelante. A estas constantes se les llaman falsas constantes pues en cualquier momento pueden reemplazarse teniendo el cuidado de hacerlo adecuadamente. El ejemplo quedaría:

$$\text{mujer}("a") \wedge \text{madre}("a", \text{luisa})$$

Donde "a" es una constante de Skolem, que puede ser reemplazada en algún momento del proceso de inferencia con el nombre de la mamá de Luisa.

- Si el \exists está dentro del dominio de un \forall

Ejemplo: representar "todo ser humano tiene una madre",

$$(\forall Y) (\text{humano}(Y) \rightarrow (\exists X) (\text{madre}(X, Y)))$$

Si se elimina el \exists reemplazando a X por una "constante de Skolem", como en el caso anterior, quedaría:

$$(\forall Y) (\text{humano}(Y) \rightarrow \text{madre}("a", Y))$$

Lo que significa que todos los humanos tienen la misma madre que es "a", lo cual es un error. Esto sucede porque el reemplazo del valor de X depende del valor de Y, por tanto en este caso no se usa una constante sino una "función de skolem" y la expresión quedaría:

$$(\forall Y) (\text{humano}(Y) \rightarrow \text{madre}(m(Y), Y)), \text{ donde}$$

En cualquier momento la función $m(Y)$ puede evaluarse y devolver el nombre de la madre de Y .

- e. Mover todos los cuantificadores universales a la izquierda de la fórmula sin cambiar su orden relativo. Esto es posible gracias a que no existe ningún conflicto entre los nombres de las variables.

$$(\forall X) (\forall Y) [p(X) \vee q(Y)]$$

- f. Convertir la matriz en una conjunción de disyunciones. Utilizando la propiedad asociativa y distributiva del conector “ \vee ”.

$$a \vee (b \vee c) \equiv (a \vee b) \vee c$$

$$(a \wedge b) \vee c \equiv (a \vee c) \wedge (b \vee c)$$

- g. Crear una cláusula por cada conjunción, puesto que para que una fbf sea cierta todas las cláusulas que se han generado a partir de ella deben ser ciertas. Cuando se trabaja con varias fbf es posible combinar el conjunto de cláusulas generadas por cada una de ellas para representar los mismos hechos de las formulas generales.
- h. Normalizar las variables que aparecen en el conjunto de cláusulas generadas en el paso anterior. Esto quiere decir que no haya dos cláusulas que hagan referencia a la misma variable

Después de aplicar todo este procedimiento a un conjunto de fbf se tiene un conjunto de cláusulas, cada una de las cuales será una disyunción de literales.

Aplicación de la regla de Resolución.

Una vez obtenidas las cláusulas, la regla de resolución dice que se tomen dos de las cláusulas⁷, que cumplan con la condición de tener un literal común pero de forma que en una cláusula la literal esté positiva y en la otra negada y, se obtenga una resolvente con la disyunción de todas las literales de las generatrices excepto la pareja de literales que fue común. Esta pareja de literales como puede ser $p(X) \vee \sim p(X)$, constituye una tautología y por tanto se puede eliminar. En cada paso únicamente se puede eliminar una pareja de literales. Durante un proceso de inferencia mediante la regla de resolución, es necesario realizar operaciones de sustitución y unificación, que permitan obtener las literales comunes en las generatrices seleccionadas, las cuales se deben referir al mismo predicado, es decir, estas literales deben tener la misma constante simbólica para representar el predicado y el mismo número y tipo de argumentos.

Sustitución: operación que se realiza para hacer coincidir argumentos de predicados. Existen dos tipos de sustituciones así:

- a. **Alfabética:** Cuando se cambia el nombre de las variables en una premisa.

Ejemplo: $(\forall X)(\sim \text{hombre}(X) \vee \text{mortal}(X))$,

Se puede hacer la sustitución de X por Y y se tendría:

$$(\forall X)(\sim \text{hombre}(Y) \vee \text{mortal}(Y))$$

- b. **Particularización de variables:** cuando se reemplaza una variable por una constante.

⁷ Estas cláusulas se llaman generatrices

Ejemplo: $(\forall X) (\sim \text{mamífero}(X) \vee \text{vertebrado}(X))$,

Podría sustituirse "X" por la constante: "ballena", y quedaría

$(\forall \text{ballena}) (\sim \text{mamífero}(\text{ballena}) \vee \text{vertebrado}(\text{ballena}))$,

Unificación⁸: cuando se tienen sentencias compuestas por predicados y conectivos lógicos, se debe evaluar la veracidad de cada uno de sus componentes para determinar si toda la sentencia es verdadera o falsa. Para ello, se busca en el conjunto de reglas la forma de establecer la veracidad de los predicados componentes. Un predicado componente se dice que es verdadero si se identifica con una regla de la base de información. En la lógica de predicados, este proceso es algo complicado ya que las sentencias pueden tener términos variables. A los predicados que tienen variables por argumentos, se los denomina **patrones**.

La **unificación** es el proceso de computar las sustituciones apropiadas que permitan determinar si dos expresiones lógicas, ya sean predicados o patrones, coinciden.

El proceso de unificación involucra los siguientes pasos:

- Todo predicado que no contenga variables en sus argumentos, deben tener una regla que se identifique totalmente, para considerarlo como verdadero.
- Si un predicado contiene una variable, ésta debe ser asociada a un valor determinado.
- El proceso de identificación continua asumiendo que el valor de la variable es el valor asociado, en cualquier lugar que ésta aparezca.

⁸ http://www.monografias.com/trabajos/iartificial/pagina4_21.htm

- Los conectivos lógicos son aplicados a todos los predicados, para determinar la veracidad de la sentencia dada.

Ejemplo:

Existen las premisas $P_1: (\forall X)(\sim \text{reptil}(X) \vee \text{vertebrado}(X))$

$P_2: (\forall Y)(\sim \text{vertebrado}(Y) \vee \text{tienen}(Y, \text{ciclo-vida-completo}))$

Las literales $\text{vertebrado}(X)$ y $\sim \text{vertebrado}(Y)$ pueden unificarse pues mediante la sustitución alfabética de X por Y , se tiene: **$\text{vertebrado}(X)$ y $\sim \text{vertebrado}(X)$** , que son literales que coinciden en el predicado y los argumentos, por tanto puede decirse que son literales comunes a las dos premisas o cláusulas

d. Resolución por Refutación: la resolución por refutación se puede definir como un proceso que toma un conjunto de cláusulas que se suponen ciertas, y, basándose en la información que las otras proporcionan, genera nuevas cláusulas que representan restricciones a la satisfacibilidad de las cláusulas originales. Surge una contradicción cuando una cláusula se vuelve tan restringida que no hay forma de hacerla verdadera. Esto se indica con la generación de la cláusula vacía.

Intenta crear una contradicción con la negación de la sentencia original, demostrando, por lo tanto, que la sentencia original es verdadera.

Ejemplo: se consideran las siguientes sentencias⁹

- cualquiera que pueda leer es letrado

$P_1: (\forall X) [r(X) \rightarrow l(X)]$

- Los delfines no son letrados

$P_2: (\forall X) [d(X) \rightarrow \sim l(X)]$

- Algún delfín es inteligente

$$P_3: (\exists X) [d(X) \wedge i(X)]$$

A partir de éstas sentencias se quiere probar la sentencia:

- Alguien que es inteligente no puede leer:

$$P_4: (\exists X) [i(X) \wedge \sim l(X)]$$

El conjunto de cláusulas correspondiente en las sentencias 1 a 3 es:

$$P_1: \sim r(X) \vee l(X)$$

$$P_2: \sim d(Y) \vee \sim l(Y)$$

$$P_{3a}: d(a)$$

$$P_{3b}: i(a)$$

Estas variables han sido normalizadas y “a” es una constante de Skolem.

La negación del teorema que debe probarse, convertida en cláusula, es:

$$P_4': \sim i(X) \vee r(Z)$$

Para probar este teorema por refutación habrá que generar resolventes del conjunto de cláusula P_1 a P_3 y P_4' , añadiendo estas resolventes al conjunto, y continuando así hasta que se produzca la cláusula vacía; una posible prueba, produce la siguiente secuencia de resolventes:

$$P_5: r(a), \text{ resolvente de } P_{3b} \text{ y } P_4'$$

$$P_6: l(a), \text{ resolvente de } P_5 \text{ y } P_1$$

$$P_7: \sim d(a), \text{ resolvente de } P_6 \text{ y } P_2$$

$$P_8: \text{NADA}, \text{ resolvente de } P_7 \text{ y } P_{3a}$$

Como se llega a una contradicción pues se llega a vacío, se puede afirmar que la Conclusión si se infiere de las premisas.

⁹ extraído de Nil. J Nilsson, Principios de Inteligencia Artificial

Estrategias de Simplificación

Estas estrategias se aplican para eliminar o simplificar cláusulas en el proceso de resolución, lo que facilita la obtención de resolventes más simplificadas. Estas simplificaciones deben ser tales que el conjunto de cláusulas simplificado sea incompatible, si y solo si, lo era el original.

Entre las simplificaciones están:

- a. **Eliminación de Tautologías:** se puede eliminar ya que cualquier conjunto incompatible que contenga tautologías seguirá siéndolo después de suprimirla.

Ejemplo:

$p(X) \vee q(X) \vee \sim q(X)$ se traduce a $p(X)$

- b. **Atribución de procedimientos:** en ocasiones es conveniente evaluar las literales de las cláusulas en lugar de incluirlas en el conjunto básico. Una literal se evalúa interpretándola ejecutando procedimientos atribuidos a ésta y si la evaluación da falso, puede eliminarse su aparición en la cláusula

Ejemplo:

$p(X) \vee q(Z) \vee \text{mayor}(X,Y)$

Si es posible atribuir a literal “mayor” un procedimiento que determina si el primer argumento es mayor que el segundo y se asigna a $X=5$ y a $Y=10$. Al evaluarse se obtiene para mayor (5,10) el valor F y, por tanto, la cláusula puede ser reemplazada por: $p(X) \vee q(Z)$, que serán las que definirán el valor verdadero de la cláusula.

- c. **Eliminación por absorción:** por definición una cláusula $\{L_i\}$, es absorbida por otra $\{M_i\}$ si existe una sustitución S tal que al aplicarla sobre $\{L_i\}$, entonces ésta se convierte en un subconjunto de $\{M_i\}$

Ejemplo:

$p(X)$ absorbida por $p(Y) \vee q(Z)$

$p(X)$ absorbida por $p(a)$

$p(X)$ absorbida por $p(a) \vee q(Z)$

$p(X) \vee q(A)$ absorbida por $p(f(A)) \vee q(A) \vee r(Y)$

En un conjunto incompatible, una cláusula que sea absorbida por otra del conjunto puede eliminarse sin que se afecte la incompatibilidad.

2.3.4 Ventajas del uso de la lógica de Predicados

- Facilidad para actualizar la base de conocimiento por la independencia de las declaraciones en la descripción del conocimiento.

2.3.5 Desventajas del uso de la Lógica de Predicados

- **Manejo de incertidumbre:** una de las mayores desventajas de la lógica de predicados es que sólo dispone de dos niveles de veracidad: verdadero y falso. En cierta forma el PROLOG ha logrado disminuir esta desventaja, permitiendo la inclusión de factores de certeza.
- No maneja tiempos verbales, requisito importante en problemas en los cuales es necesario su control.
- Para una misma expresión en lenguaje natural hay muchas representaciones en lógica y viceversa.
- En ocasiones se presta para interpretaciones ambiguas.

2.4 REGLAS DE PRODUCCIÓN

2.4.1 Definición

Las reglas de producción son una frc (forma de representación del conocimiento) que permite la representación del conocimiento en la estructura:

Si condición(es) —→ antecedente
ENTONCES conclusión —→ **consecuente**

- a. La parte **SI (IF)**, es el antecedente, premisa, condición o situación que debe cumplirse. Las condiciones pueden estar ligadas por conectores tales como "and" u "or".
- b. La parte **ENTONCES (THEN)**, es el consecuente, conclusión, acción o respuesta. en conclusión son las acciones a realizar al usar la regla. Éstas acciones pueden ser entrar condiciones de nuevos estados de las base de datos, modificar las descripciones existentes, dar respuestas al usuario.

Ejemplo:

si esqueleto = òseo **and**
reproductor = vivíparo **and**
respiración = pulmonar **and**
temperatura-sangre = caliente
entonces clasificación = mamífero

Esta forma de Representación de Conocimiento es la más utilizada, ya que facilita procesos de inferencia y ha sido usada con éxito en la implementación de herramientas en sistemas expertos comerciales. El uso de las Reglas de

El proceso de solución de un problema en los sistemas basados en reglas va realizando una serie de inferencias que crean un sendero entre la definición del problema y su solución. Las inferencias están concatenadas y se realiza en forma progresiva, por lo que se dice que el proceso de solución origina una **cadena de inferencias** para cuya construcción se aplican estrategias de búsqueda.

2.4.3 Estrategias de Búsqueda

Las estrategias de búsqueda utilizadas en los sistemas basados en reglas son:

a. Razonamiento Progresivo o encadenamiento hacia adelante

Se comienza a partir de un conjunto de datos colectados a través de observación y se evoluciona hacia una conclusión. Se examina cada una de las reglas para verificar si los datos observados satisfacen las premisas o condiciones de algunas de las reglas. Si una regla es exitosa es ejecutada derivando nuevos hechos que pueden ser utilizados por otras reglas para derivar hechos adicionales y su uso progresivo genera la conclusión dentro de la base de datos. El proceso de examinar reglas para verificar si se satisfacen las premisas se denomina ***interpretación de reglas***.

Los aspectos del proceso a tener en cuenta en este tipo de encadenamiento, son:

- 1. Unificación:** en las reglas se toman los hechos (datos), para ver cuáles son las reglas cuyas condiciones resultan satisfechas por los datos. Para decir que una regla ha sido satisfecha, se requiere que todas las premisas o antecedentes de la regla sean verdaderas o se conozcan.

2. Resolución de Conflictos: Si durante el proceso aparecen varias reglas que puedan elegirse para la interpretación, se requiere de una estrategia de control para hacer la selección.

Algunas estrategias de control pueden ser:

- ❑ Escoger la primera que unifique todos los datos.
- ❑ Escoger la primera regla que pueda seleccionarse.
- ❑ Establecer un criterio de actualidad donde las reglas que primero son ejecutadas, son las que contienen la información mas reciente.
- ❑ Usar una función heurística que evalúe la bondad de la regla para llegar a la solución, es decir, que mida que tan conveniente es irse por ese camino para obtener la solución.
- ❑ Seleccionar la reglas mas completa o la que tenga mas condiciones, pues la mayoría de las veces estas reglas dan información más completa o abarcan conocimiento más complejo.

3. Ejecución.- El último paso en la interpretación de reglas es la ejecución de la regla. La ejecución puede dar lugar a varios resultados posibles:

- Nuevo hecho (o hechos) pueden ser derivados y añadidos a la base de hechos, o
- Nueva regla (o reglas) pueden ser añadidas al conjunto de reglas (base de conocimiento) que el sistema considera para ejecución.
- Respuestas para el usuario.

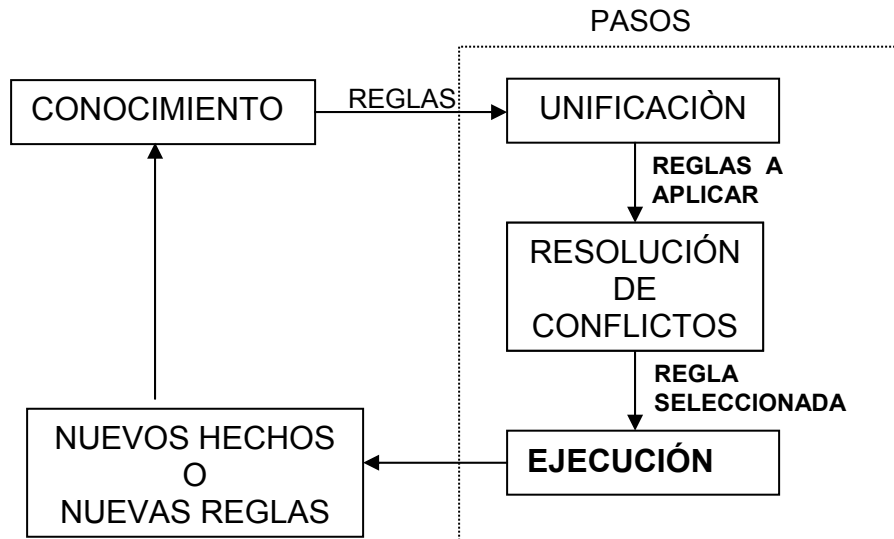


Figura 1. Proceso de razonamiento progresivo

De esta forma, la ejecución de las reglas procede de una manera progresiva es decir, desde las condiciones hacia las conclusiones, desde los datos a los objetivos finales.

Ejemplo:

Se tiene la siguiente base de reglas:

- R₁: **si a y b entonces c**
- R₂: **si c entonces d**
- R₃: **si d y a entonces f**
- R₄: **si f y h entonces x**
- R₅: **si f entonces h**
- R₆: **si c y a entonces h**

En la base de datos se tiene que *a* y *b* son datos ciertos

Meta: es probar *X*.

Solución:

Como se tienen dos datos que son a y b se busca una regla cuyas condiciones satisfagan la información. Para el primer paso, se elige R_1 porque sus condiciones son satisfechas por la base de datos. Por consiguiente, usada la regla en la base de datos se almacenarían a , b y c .

	<i>Base de Datos</i>	<i>Reglas</i>
1	a y b	R_1
2	C	

Para el segundo paso, como se tiene a , b y c se observa que hay un conflicto pues es posible elegir las reglas 2 y 6. Usando la estrategia de control de elegir la primera regla que aparezca, se selecciona R_2 y así se sigue el proceso, que se comporta como se muestra a continuación.

	<i>Base de Datos</i>	<i>Regla elegida</i>	<i>Regla candidata</i>
1	a y b	R_1	R_1
2	C	R_2	R_2 y R_6
3	D	R_3	R_3 y R_6
4	F	R_5	R_5 y R_6
5	H	R_4	R_4 y R_6
6	X	SOLUCIÓN	

- **Razonamiento regresivo o encadenamiento hacia atrás.**

Este razonamiento comienza con la conclusión deseada y verifica si los hechos que existen pueden dar lugar a la obtención de un valor para esta conclusión o se busca una regla donde la conclusión sea la condición cuya validez se está

probando y se sigue trabajando hacia atrás, hasta llegar a la confirmación de la validez de las condiciones que están en la base de datos.

Este método se puede realizar por dos caminos:

1) **Retroceso:** este tipo de búsqueda se lleva a cabo cuando

- ❖ Hay un punto en donde las reglas no pueden probar la validez de una condición.
- ❖ Se llega a un número definido de reglas para la búsqueda, o sea, se llega a un nivel de profundidad.
- ❖ Se llega a un hecho o hechos a los que ya se ha llegado, utilizando otros caminos.

2) **Ciclaje:** se presenta cuando el objetivo a probar aparece en una cadena de cláusulas las cuales tratan de probarla.

Ejemplo: R_1 : si X and Y entonces Z
 R_2 : si Z and W entonces X

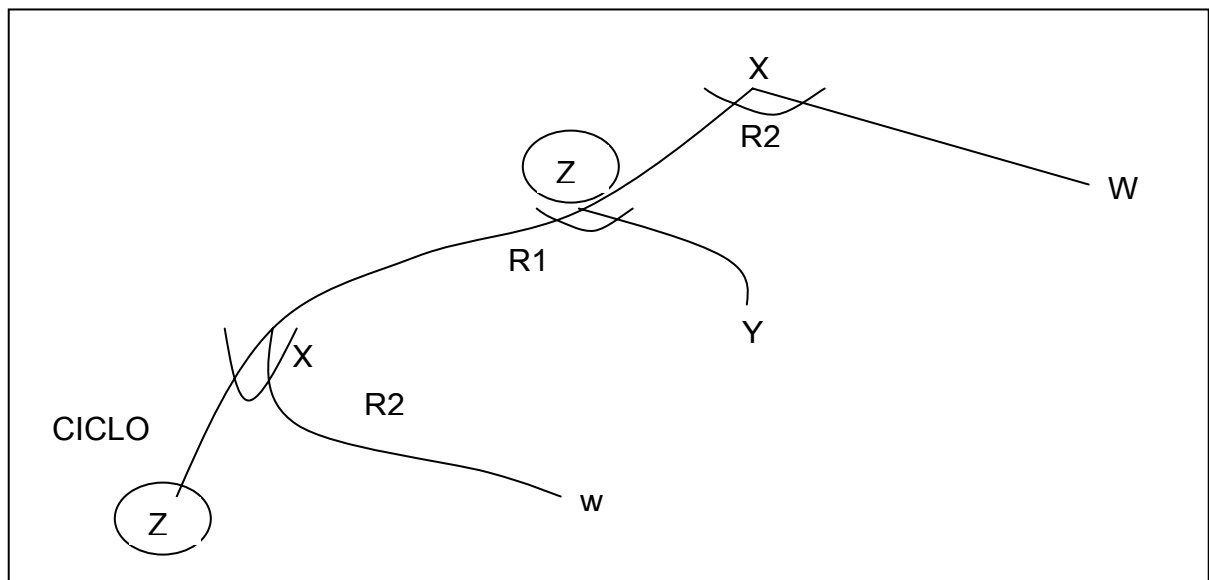


Figura 2. Ciclaje en Búsqueda hacia atrás

En el encadenamiento hacia atrás también se presentan conflictos, pues en ocasiones es posible utilizar varias reglas durante el proceso. Para resolver el conflicto se utilizan las mismas estrategias de control enunciadas en el encadenamiento hacia delante. Algunas de ellas son:

- Usar la primera regla que unifique.
- Usar la regla más completa, o sea, la que tenga más condiciones. .
- Usar la regla que tenga una mejor función de evaluación.

- **Búsqueda en Extensión**

Este tipo de búsqueda permite buscar soluciones explorando al mismo tiempo todos los posibles caminos, evaluando en cada nivel si se ha tenido o no, éxito.

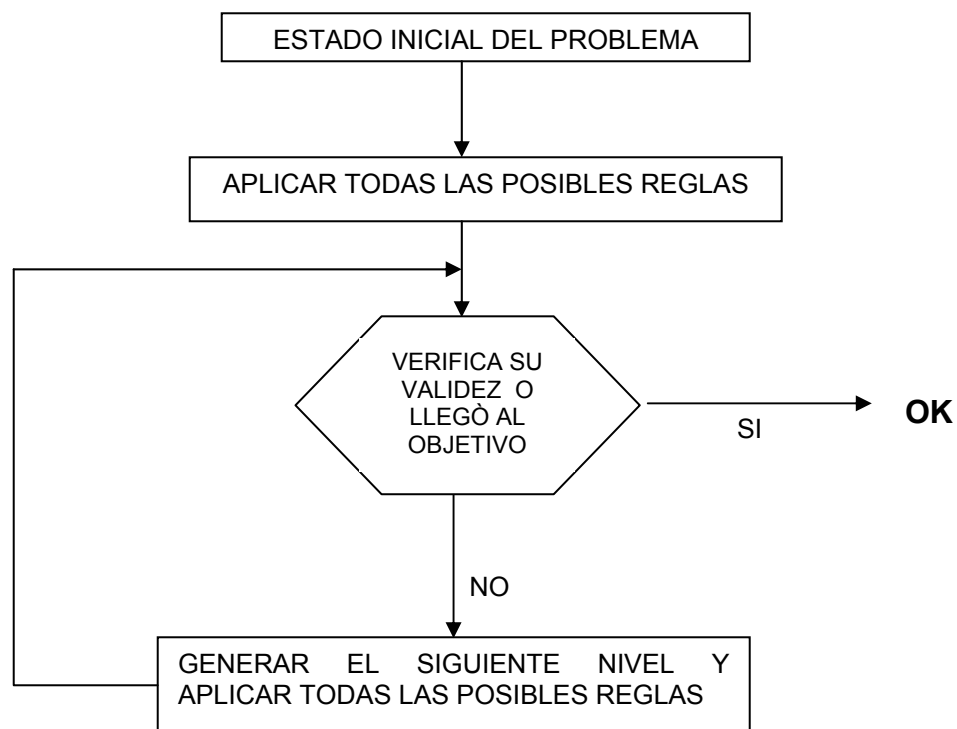


Figura 3. Extensión

- **Búsqueda en Profundidad**

Es la búsqueda que selecciona una regla determinada y explora por ese camino hasta llegar al objetivo o hasta que se falle el proceso.

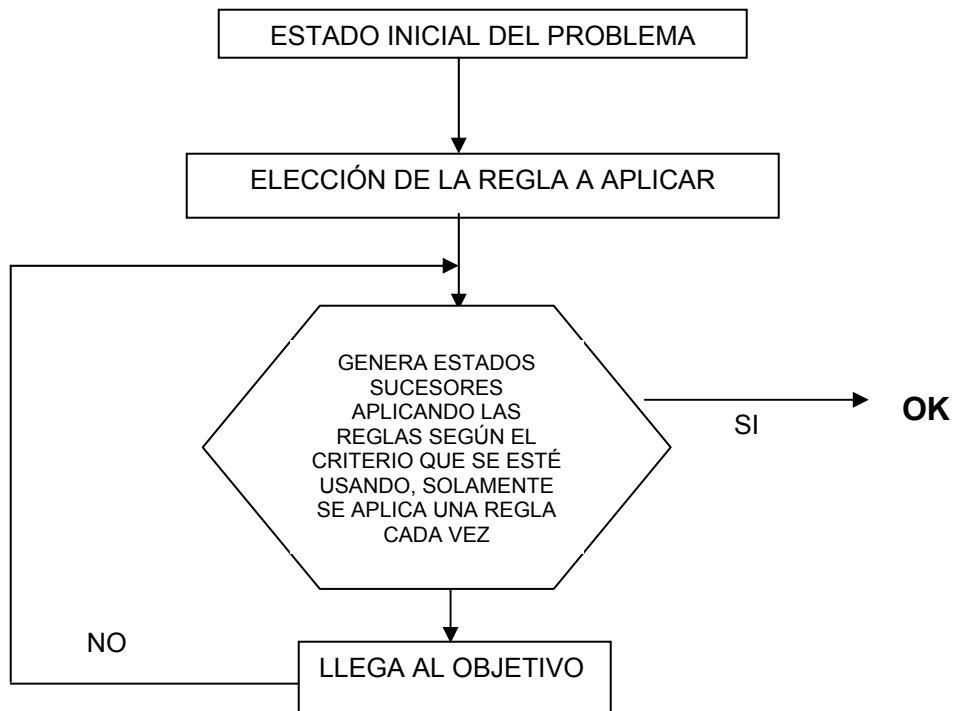


Figura 4. Profundidad

- **Búsqueda Heurística y a Ciegas.**

Otra forma de resolver problemas relacionados con reglas de producción es utilizando estrategias de búsqueda heurística y a ciegas.

Búsqueda Heurística: es la búsqueda que utiliza información del dominio del problema. Se utilizan funciones de evaluación de la bondad de las reglas, que permiten elegir aquella regla con mayor probabilidad de éxito para encontrar la solución.

Búsqueda a Ciegas: es la búsqueda que no utiliza información para la selección de las reglas, por lo se escoge la primera regla que unifique o se utilizan todas.

2.4.4 Ventajas de su Uso

- Facilidad en adición y modificación del conocimiento.
- Las reglas de decisión están escritas casi en lenguaje natural.

2.4.5 Desventajas de su Uso

- Encadenamiento infinito.
- Incorporación de conocimiento nuevo contradictorio.
- Modificación de reglas existentes.

2.5 REDES SEMÁNTICAS

2.5.1 Definición

Las **redes semánticas**, fueron originalmente desarrolladas para representar el significado o semántica de oraciones en inglés, en términos de objetos y relaciones.

Actualmente, el término **redes asociativas** es más ampliamente utilizado para describirlas ya que no sólo se las usa para representar relaciones semánticas, sino también para representar asociaciones físicas o causales entre varios conceptos u objetos. Son muy apropiadas para representar conocimiento de naturaleza jerárquica. Su concepción se basa en la asociación de conocimientos que realiza la memoria humana. Las principales aplicaciones son: comprensión de

lenguaje natural, bases de datos deductivas, visión por computadora y sistemas de aprendizaje.

Las redes semánticas se caracterizan por representar el conocimiento en forma gráfica, su principal idea es que el significado de un concepto depende del modo en que se encuentre conectado con otros conceptos.

2.5.2 Componentes

En su construcción se destacan los siguientes componentes:

- a. **NODOS:** se denominan también objetos (elementos del conocimiento). Y pueden ser:
 - **Objetos físicos**, se refieren a todo objeto que pueda verse y tocarse.
 - **Objetos conceptuales**, hace alusión a hechos, eventos, cualidades.
 - **Objetos descriptivos**, permiten asociar los nodos con los enlaces.

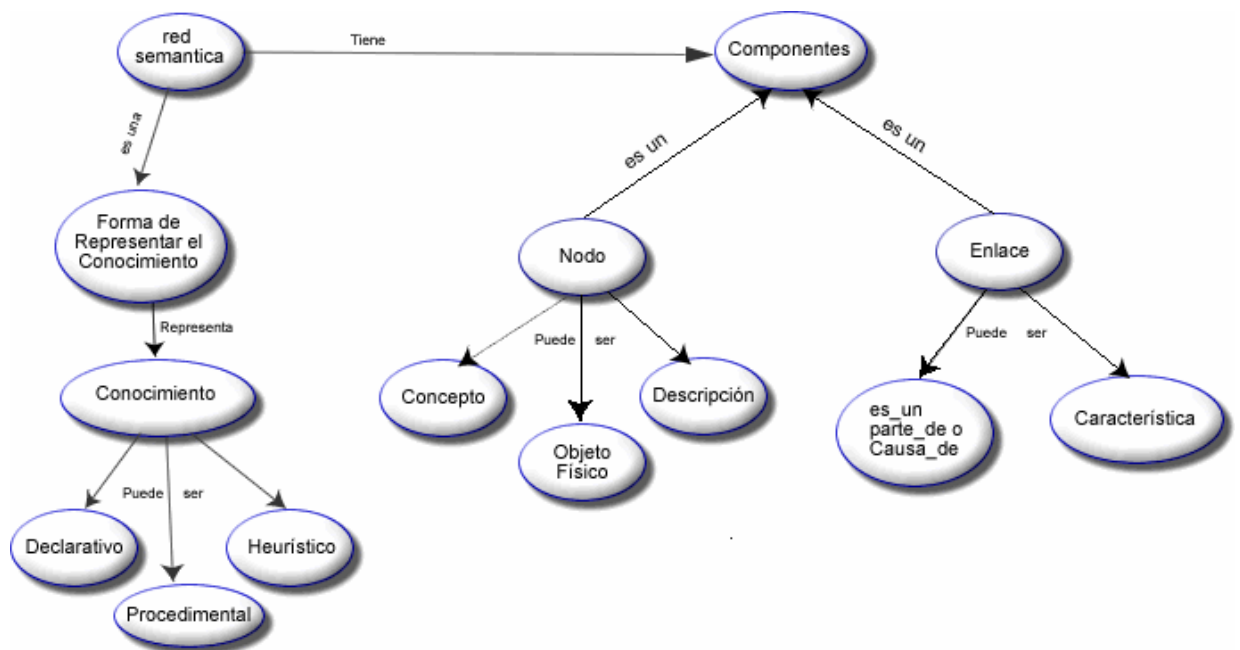


Figura 5. Concepto de Red Semántica

- b. ENLACES: permiten definir relaciones que se presentan entre los objetos del dominio, como por ejemplo la de un elemento con su clase (es_un), la parte con su todo (parte_de) y causa con su síntoma (causa_de) y, además, pueden representar definiciones o características.

2.5.3 Reglas de Búsqueda

Dos formas de uso para la interpretación del lenguaje con Redes Semánticas son: *encontrar las similitudes y diferencias entre palabras, y expresar oraciones congruentes con base en la información en la red.*

En general se puede usar este tipo de estructuras para diferentes tipos de razonamiento:

- b. **Búsqueda asociativa:** encontrar si están relacionados dos o mas conceptos, y su tipo de relación mediante el seguimiento de la red hasta encontrar las interacciones.
- b. **Reconocimiento:** dada una serie de características, encontrar el concepto que mejor las define mediante su búsqueda y seguimiento en la red.
- c. **Descripción:** expresar un concepto con base en sus componentes y relaciones entre ellas (lenguaje natural).
- d. **Herencia:** la gran potencia de las redes semánticas está en que permiten el manejo de la herencia de propiedades entre clases, subclases y elementos a través de los enlaces es_un o es_una, de forma que no se presente redundancia de conocimiento.

La herencia de propiedades, característica potente de las redes, permite que cualquier propiedad que se declare como verdadera para una clase de objetos será cierta para cualquier subclase, a menos que se manejen excepciones.

Ejemplo:

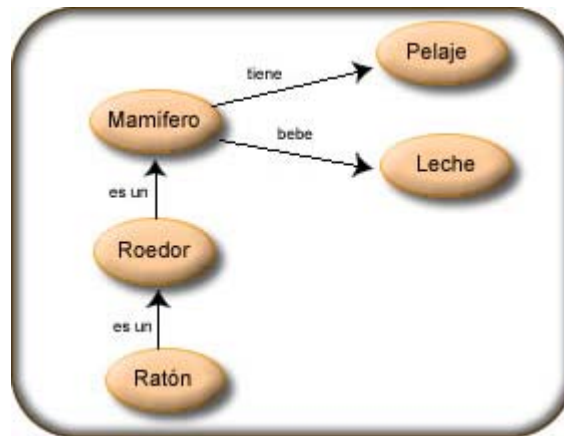


Figura 6. Ejemplo de Red Semántica

En el anterior ejemplo el nodo ratón hereda aquellas característica o enlaces que no tiene definidas y que poseen sus clases superiores que son roedor y mamífero, por lo que quedaría con las características de bebe leche y tiene pelaje. Al nodo mamífero se le llama **nodo padre**, pues a partir de él se genera el resto de la red semántica.

e. Confrontación: es la operación que se realiza cuando se trata de responder a consultas del usuario y se usa el conocimiento representado en la red semántica.

Ejemplo [Tomado y modificado de Fernández Severino y otros, 1998]:

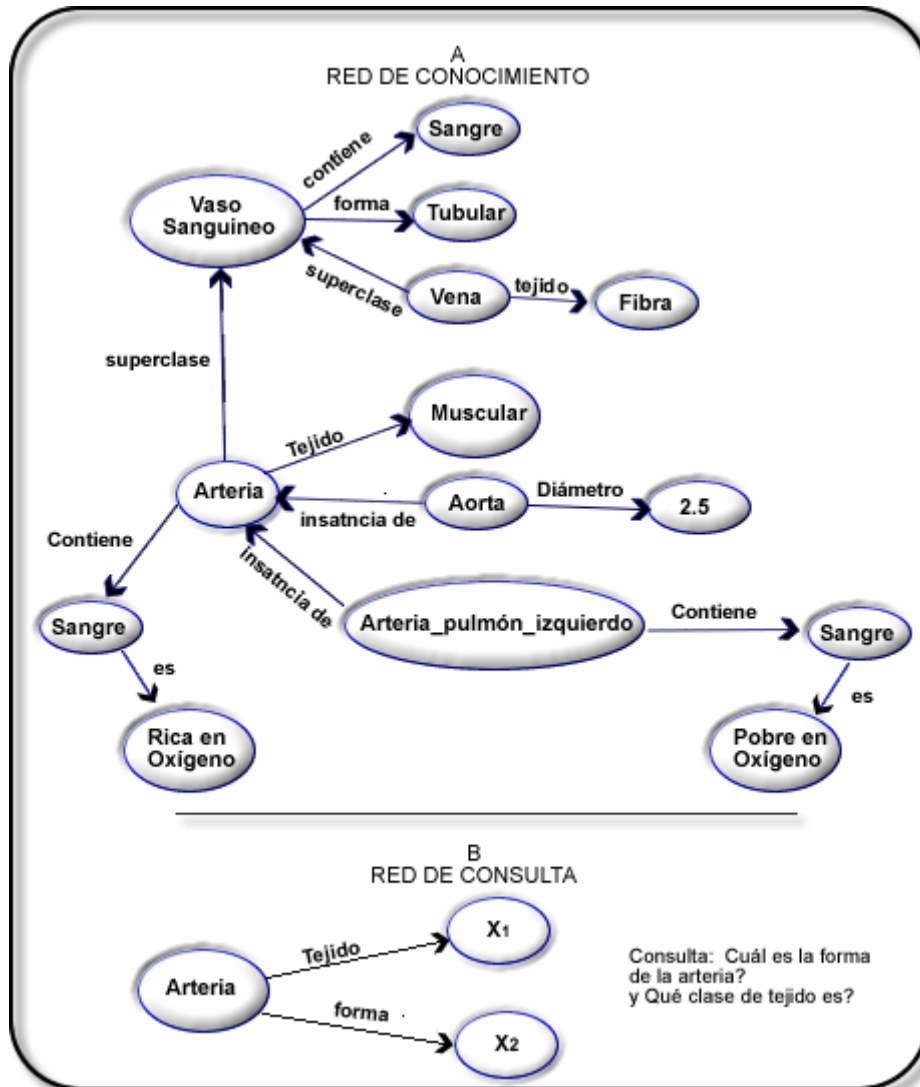


Figura 7. Operación de Confrontación en redes

El nodo arteria de la red objetivo coincide con el nodo arteria de la red de conocimiento, de igual forma también coincide el enlace tejido y el enlace forma que se hereda de su clase superior vaso sanguíneo, con la red de conocimiento. Haciendo la operación de confrontación, la variable **X1** toma el valor de muscular y la variable **X2** el valor de tubular.

2.5.4 Ventajas de las Redes Semánticas

Las redes semánticas tienen dos ventajas sobre los sistemas basados en reglas y sobre los basados en lógica:

- a. Permiten la declaración en forma explícita de importantes asociaciones.
- b. Debido a que los nodos relacionados están directamente conectados, y no se expresan las relaciones en una gran base de datos, el tiempo que toma el proceso de búsqueda por hechos particulares puede ser significativamente reducido.

2.5.5 Desventajas de las Redes Semánticas

- a. No existe una interpretación normalizada para el conocimiento expresado por la red. La interpretación de la red depende exclusivamente de los programas que manipulan la misma.
- b. La dificultad de interpretación a menudo puede derivar en inferencias inválidas del conocimiento contenido en la red.
- c. La exploración de una red asociativa puede derivar en una explosión combinatoria del número de relaciones que deben ser examinadas para comprobar una relación.

2.6 FRAMES

2.6.1 Definición

Un frame (marco) es una estructura de datos apropiada mediante la cual se describe características y relaciones entre objetos de un dominio. Los frames organizan el conocimiento en objetos y eventos que resultan apropiados para situaciones específicas. Igualmente, un frame representa un objeto o situación describiendo la colección de atributos que posee.

2.6.2 Componentes

Un frame consta de:

Nombre: debe ser único para diferenciarlo de los demás.

Padre: Equivale a la clase superior a la cual pertenece el objeto que describe el frame y es la que le permite el mecanismo de herencia.

Ranuras o Slots: representan una propiedad estándar o atributo de la clase u objeto representado. Cada slot puede contener uno o más **enlaces** (facets). Cada *enlace* tiene un valor asociado. Es posible definir diversos *enlaces* para cada *ranura*, por ejemplo:

- *Rango:* el conjunto de posibles valores para la ranura.
- *Valor:* el valor de la ranura, puede ser numérico, símbolos, funciones para evaluar o llamadas a programas que se ejecutan en el procesamiento.
- *Default:* el valor a ser asumido si no se especifica alguno.
- *Multivaluado:* especifica si el campo admite o no varios valores simultáneamente.
- *Certeza:* hace referencia a la credibilidad del valor asignado al campo.
- *Facetas de Interfaz:* contiene información destinada a la interacción con el usuario.

Además los *enlaces* pueden ser procedimientos que residen en la base de datos y están aguardando para ser utilizados cuando se les necesite. Entre los más comunes se pueden mencionar:

- *Si-Necesitado*: Procedimiento(s) para determinar el valor actual de una ranura, y éste no tiene ningún valor asignado.
- *Si-Agregado*.- Procedimiento(s) a ejecutarse cuando un valor es especificado para una ranura.
- *Si-Modificado*.- Procedimiento(s) a ejecutarse si el valor de una ranura es cambiado.
- *Si-Acceso*: *procedimiento(s)* a ejecutarse cuando se solicita el valor de un campo, aunque ya tenga un valor asignado.
- *Si-Borrado*: *procedimiento(s)* a ejecutarse al ser eliminado el valor de un campo.

A estos procedimientos también se les denomina *demons* y representan la habilidad de combinar conocimiento procedimental dentro de la estructura de conocimiento declarativo del frame.

El esquema del frame sería:

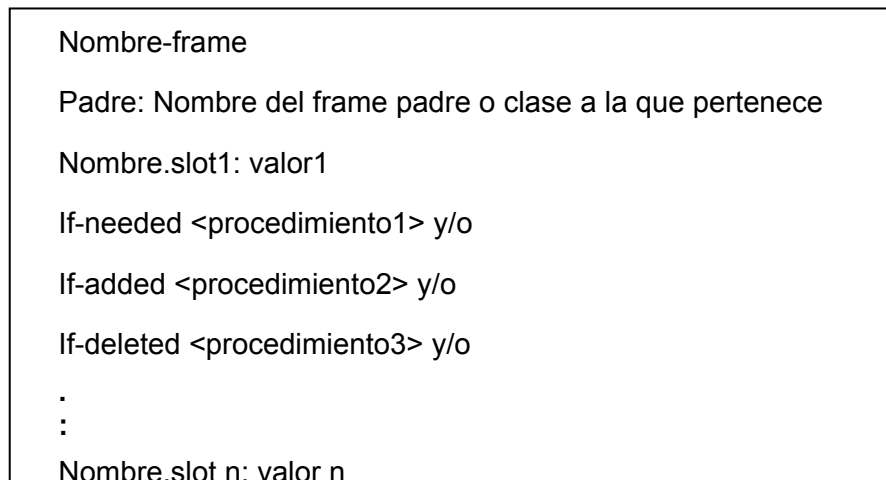


Figura 8. Esquema de un Frame

2.6.3 Reglas de Búsqueda

Como las redes semánticas, los frames se pueden utilizar para diferentes clases de inferencias:

- 1) **Reconocimiento:** dados ciertos atributos de un objeto (campos) encontrar el objeto, o el mas similar, el proceso para recuperar valores de los slots es el siguiente:
 - (a) Encontrar el frame referenciado y, dentro de él, el slot específico, si se encuentra el frame con el slot referenciado sigue el paso (c), y si no tienen el slot va al paso (b).
 - (b) Verificar si el padre del frame referenciado en la consulta o un antepasado suyo posee el respectivo slot. Si se encuentra, por herencia se le añade el slot y/o valor, al frame de la consulta. Si no hay éxito, se informa la imposibilidad de obtener un valor y si lo hay se sigue con el paso (c).
 - (c) Si el slot del frame referenciado tiene como propio o heredado el procedimiento if-needed, éste se ejecuta y si tiene éxito, se recupera el valor del slot para el usuario, de lo contrario no se obtendrá el valor. Si el slot no tiene o no hereda el procedimiento if-needed sencillamente se recupera su valor para quien hace la consulta.

Para añadir valores a los slots se procede de la siguiente forma:

- (a). Encontrar el frame referenciado y el slot respectivo, si no se encuentra no habrá respuesta, y si lo encuentra con el slot sigue (c), si no a (b).
- (b) Verificar si el frame padre o antepasado posee el slot respectivo. Si se encuentra, el frame referenciado hereda el slot solamente y va a (c), si no se encuentra, el frame no hereda el slot y se termina el proceso.

- (c) Verificar si el slot referenciado o el del padre, tiene adjunto el procedimiento if-added, si lo tiene se ejecuta su procedimiento y si tiene éxito se le asigna al slot el valor proporcionado por el usuario. Si el slot no tiene o no hereda este procedimiento sencillamente se asigna el valor respectivo al slot.

Para borrar valores a los slots se procede así:

Es semejante al proceso de obtención de valores de slots, salvo que la operación que se hace es borrar el valor del slot, luego el paso (c) quedaría así:

- (c) Verifique si el slot del frame referenciado tiene como propio o heredado el procedimiento if-needed, éste se ejecuta y si tiene éxito, se borra el valor del slot . Si el slot no tiene o no hereda este procedimiento sencillamente se borra el valor.

- 2) **Razonamiento por valores típicos (default reasoning):** dada cierta información incompleta de un objeto, deducir la información faltante.
- 3) **Herencia:** Dada cierta información de un objeto, deducir los atributos faltantes de su(s) superclase, cada marco hereda los campos de sus antepasados en la red.

Ejemplo:

Los reptiles son vertebrados de vida terrestre y cuerpo con escamas

Al organizar la información en un frame quedaría así:

Vertebrados	Tipo: animal
	Esqueleto: si
Reptil	Tipo: vertebrado
	Medio_vida: tierra
	Piel: escamada

2.6.4 Ventajas de los Frames

Las ventajas que se pueden establecer para los sistemas basados en frames son las siguientes:

- Facilidad de proceso guiado por las expectativas. Un sistema basado en frames, mediante los *demons* es capaz de especificar acciones que deben tener lugar cuando ciertas condiciones se han cumplido durante el procesamiento de la información.
- El conocimiento que posee un sistema basado en frames es significativamente más estructurado y organizado, que el conocimiento dentro de una red semántica o en una base de conocimientos que utilice lógica de predicados o reglas de producción.
- Los frames pueden ser estructurados de forma que sean capaces de determinar su propia aplicabilidad en determinadas situaciones. En el caso de que un frame en particular no sea aplicable, puede sugerir otros frames que pueden ser apropiadas para la situación.
- Se puede fácilmente almacenar en las ranuras valores dinámicos de variables, durante la ejecución de un sistema basado en conocimiento. Esto puede ser

particularmente útil para aplicaciones de simulación, planeamiento, diagnóstico de problemas o interfaces para bases de datos.

2.6.5 Desventajas de los Frames

Las principales desventajas que se pueden establecer para la representación del conocimiento mediante frames, son:

- Dificultad de representar objetos que se alejen considerablemente de estereotipos.
- No tiene la posibilidad de acomodarse a situaciones u objetos nuevos.
- Dificultad para describir conocimiento heurístico que es mucho más fácilmente representado mediante reglas.

2.7 EDUCACIÓN E INFORMATICA

El computador es una herramienta valiosa que busca facilitar la vida cotidiana y se ha vuelto imprescindible en diferentes áreas, especialmente en la educación, como instrumento que ofrece ventajas útiles para el procesamiento automático de la información, y que exige al docente el desempeño de un rol dinámico, puesto que las experiencias educativas en línea requieren de un acompañamiento y actualización constantes; que permita convertir esta herramienta en un verdadero apoyo para los procesos de enseñanza y aprendizaje, ya que su efectividad radica tanto en las características técnicas que posee la máquina, como en los desarrollos de la tecnología educativa en que se fundamente la utilización que de éste hacen los participantes del proceso.

Las nuevas Tecnologías de la Información y Comunicación (TIC) son definidas por los expertos como “tecnología disruptiva” allí donde se ha aplicado. Rompen e

innovan, pero reforzando lo fundamental en cada ámbito. Así ha sucedido, por ejemplo, en las finanzas, en el comercio electrónico o en la comunicación interpersonal.

También en educación las TIC rompen esquemas o didácticas tradicionales, pero reforzando los aspectos básicos de la enseñanza y del aprendizaje. ¿Qué aspectos? Primero, la información, permanentemente disponible y fácilmente accesible, con la que se construye el conocimiento. Segundo, la actividad o intervención permanente del alumno; el programa útil educativamente es el que hace a la máquina “tonta” para que el alumno actúe inteligentemente. Y, finalmente, el más importante, la interacción.

Las TIC son una herramienta que potencia la intervención del profesor: la comunicación con los alumnos y la dirección del aprendizaje. Sin esta interacción del profesor con el alumno la información puede ser simple ruido y la actividad del alumno corre el riesgo de ir a ciegas o resultar inútil. Es en la interacción donde se hila la información y se teje el conocimiento. Cambian los telares, pero lo básico de la relación entre maestro y aprendices para generar saberes y destrezas permanece: la guía y la comunicación.

2.7.1 EDUCACIÓN VIRTUAL

La educación virtual es una experiencia reciente que ha evolucionado y se ha expandido considerablemente a nivel mundial, favoreciendo el proceso de aprendizaje, gracias a las nuevas tecnologías de la informática y la comunicación (TIC), para apoyar en el estudiante el acceso y la organización adecuada de la información, orientando actividades y sugiriendo estrategias que permitan al estudiante recibir, organizar y trabajar suficiente información con significado de manera que pueda activar los conocimientos que ya tiene,

establecer relaciones y reelaborar conceptos, resolver problemas o tomar decisiones.

La educación respaldada con la tecnología posibilita el apoyo de experiencias en los cursos presenciales de los programas de pregrado y postgrado, así como de los cursos de los programas a distancia, sin embargo, su implementación exige compromisos y cambios a nivel pedagógico y organizativo, además del aseguramiento de una infraestructura que garantice una capacitación adecuada, completa y sistemática de los actores del proceso y la disponibilidad permanente de los recursos.

Como apoyo al uso de las TIC en los procesos educativos se ofrecen las experiencias de enseñanza virtual. En esta dirección “Aula Virtu@l” pretende facilitar el desarrollo de ambientes virtuales de aprendizaje. Las experiencias que se facilitan mediante los servicios de “Aula Virtu@l” consideran: el uso de las TIC como instancias de mediación; la resolución de problemas y el aprendizaje colaborativo como estrategias para favorecer el aprendizaje significativo, pues apoyan el desarrollo de competencias cognitivas, actitudinales y procedimentales; el estudiante como protagonista del aprendizaje y el profesor como facilitador o guía, desde el lado, de las experiencias de aprendizaje, y la caracterización de las experiencias virtuales como apoyo al modelo pedagógico donde se utilizan las estrategias de aprendizaje señaladas¹⁰.

¹⁰ **CORREDOR** Montagut, Marta Vitalia y otros. Aula Virtu@l: “una alternativa en educación superior”. Bucaramanga. Ediciones UIS. 2004.

2.8 INGENIERIA DEL SOFTWARE

La ingeniería del software es una tecnología multicapa. Los cimientos que son la base de la ingeniería del software están orientados hacia la calidad, el fundamento de la ingeniería del software es la capa proceso¹¹.

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad" [Jacobson 1998]. El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo".

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición.

- La concepción define el alcance del proyecto y desarrolla un caso de negocio.
- La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura.
- La construcción crea el producto y la transición transfiere el producto a los usuarios.

¹¹ **PRESSMAN**, Roger S. Ingeniería del Software un enfoque práctico. Mexico; Editorial McGraw-Hill, 1998

Los métodos de la ingeniería del software indican cómo construir técnicamente el software. Estos abarcan tareas que incluyen análisis de requisitos, diseño, construcción del programa, pruebas y mantenimiento.

Las herramientas de la ingeniería del software proporcionan un soporte automático o semi-automático para el proceso y para los métodos. Cuando se integran herramientas para que la información creada por alguna de éstas la pueda utilizar otra, se establece un sistema de soporte para el desarrollo del software llamado ingeniería del software asistida por computadora (Computer-aided software engineering CASE).

2.8.1 Prototipado Evolutivo

El prototipado evolutivo comienza con un análisis general de los requerimientos para crear un prototipo inicial. La experiencia del desarrollo del prototipo y su evaluación deben permitir la definición de especificaciones más completas y seguras para el producto definitivo. Para lograr esto, lo primero que hay que hacer es realizar una especificación del sistema, a partir de los requisitos globales que se pueden conocer inicialmente. En este caso no es necesario realizar una definición completa de los requisitos. Luego se procede a realizar un diseño rápido a partir del cual se construye un prototipo que será evaluado por el cliente quien suministra una realimentación que facilitará una validación y especificación más exacta de lo que el cliente desea, con lo que es posible refinar los requisitos para diseñar y construir el siguiente prototipo tomando como base los diseños y el código que ya se tiene.

La Figura 9, describe un esquema de flujo de trabajo a través de las fases que se ejecutan para obtener un producto final. El proceso parte de un concepto inicial donde se definen el análisis y especificaciones de los requerimientos iniciales.

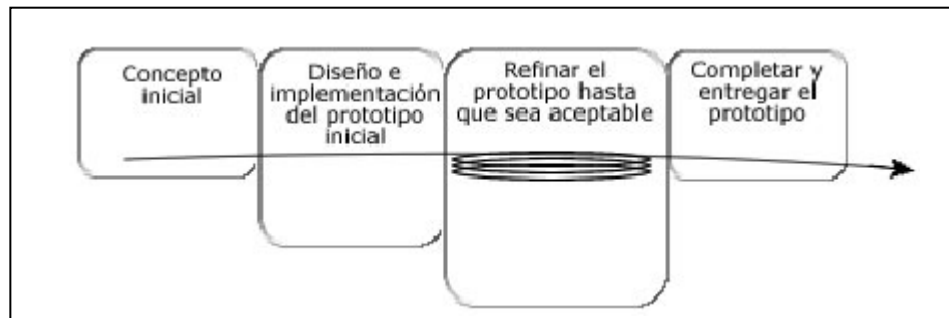


Figura 9. Prototipado Evolutivo

Características

- No exige una fuerte planificación.
- Funciona con incrementos, los cuales arrojan como resultado un prototipo, para luego agregar nuevas funcionalidades hasta alcanzar lo que el cliente desea.
- Se adapta a la mayoría de las solicitudes de modificación de los clientes, a diferencia del modelo de entrega por etapas.
- Genera signos visibles de progreso.
- No es tan complicado ni requiere de tanta sofisticación como otros modelos, por ejemplo el modelo de desarrollo espiral.

2.8.2 Lenguaje de Modelado Unificado (UML)

Constituye un estándar para el modelado de proyectos, independiente del lenguaje de programación que utilice y de las características del software; ya que UML ha sido diseñado para modelar cualquier tipo de proyecto en su etapa de diseño. En conclusión, es un lenguaje gráfico para visualizar, especificar, documentar cada una de las partes que comprende el desarrollo de software.

Lo que UML permite es crear un modelo dinámico a partir de ciertos elementos y ciertas técnicas, que representan el sistema. Los modelos creados hacen uso de notación gráfica que representa principalmente la información, los procesos y el comportamiento del sistema.

El rol de los modelos se puede sintetizar en:

- Ayudar a entender la información, la función y el comportamiento del sistema, haciendo por tanto más fácil y sistemática la tarea de analizar los requerimientos del cliente.
- Convertirse en el punto de comparación entre lo logrado y lo planificado.
- Fundamentar el diseño, proporcionando una representación lógica, de la implementación del sistema.

Objetivos de UML

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no intenta ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. Incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo y dinámico, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.

- Ser un lenguaje universal y estándar, como cualquier lenguaje de propósito general.

Diagramas en UML

Un producto software ofrece un conjunto de modelos que permite expresar la herramienta desde varias perspectivas de interés, ya sea del desarrollador como del cliente. Un ejemplo puede ser el mismo código fuente del sistema que se considera un modelo muy detallado del software y, además, también el ejecutable. Sin embargo no solo el código es suficiente para modelar un producto software; se requiere de otros modelos que lo complete desde otro punto de vista del sistema. Los elementos de UML se muestran por medio de diagramas que representan varias vistas del sistema y el conjunto de las vistas son los modelos.

Diagrama de Casos de Uso

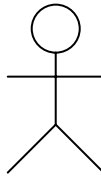
El diagrama de Casos de Uso muestra las distintas operaciones del sistema y cómo se relaciona con su entorno, ya sea con los usuarios o con otras aplicaciones. En conclusión, estos Diagramas Representan la funcionalidad que ofrece el sistema en lo relacionado a su interacción externa.

Es una herramienta esencial para la captura de requerimientos ya que su principal funcionalidad es documentar los requisitos del sistema y, además, es esencial para la planificación y el control de un proyecto interactivo. Donde el conjunto de casos de uso representa la totalidad de operaciones desarrolladas por el sistema.

Los diferentes elementos que componen un diagrama de casos de uso son los siguientes:

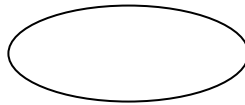
Elementos:

Actor:



Es el encargado de realizar algún tipo de interacción con el sistema. Es el rol que un usuario hace con respecto al sistema.

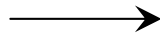
Caso de uso:



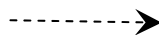
Expresa una funcionalidad del sistema. Muestra una tarea específica que lleva a cabo el actor. Se representa con una elipse con el nombre del caso del uso donde este nombre debe reflejar la tarea específica que el actor desea llevar a cabo usando la aplicación o sistema.

Relaciones: en un diagrama de casos de uso, pueden existir relaciones entre actores y casos de usos o entre varios casos de usos; es por esto que existen diversas clases de relaciones:

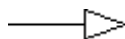
- Asociación: relación que hay entre un actor o caso de uso con otra operación o caso de uso



- Dependencia o instanciación: es una relación entre clases, en la cual una clase depende de otra. Una clase es la unidad básica que encapsula toda la información de un objeto (instancia de una clase).



- Generalización: está orientado exclusivamente para casos de uso y no para actores. Puede ser de **Uso** (<<uses>>) o de **Herencia** (<<extends>>).



extends: Se utiliza cuando se encuentran dos casos de uso similares en sus características.

uses: Se utiliza cuando se tiene un conjunto de característica que son similares en más de un caso de uso.

2.9 TECNOLOGÍAS WEB

2.9.1 HTML

HTML (hypertext markup language) es un lenguaje utilizado para desarrollar páginas y documentos Web y está compuesto por dos partes: la que se muestra en la pantalla (texto, imágenes, enlaces, etc.) y los códigos (formado por etiquetas o tags), transparentes al usuario y son los encargados de indicar cómo se debe mostrar la información en la pantalla.

Para editar o crear una página HTML y posteriormente visualizarla, se requiere de un editor de texto ASCII (donde se incluyen las etiquetas necesarias para la presentación de la página) y un explorador Web como Internet Explorer o Netscape. Existen diversos editores de texto de HTML que van desde el Bloc de notas hasta otros mas específicos que insertan automáticamente las etiquetas correspondientes al formato del texto que se escriba o a las imágenes que se inserten. Posteriormente, las páginas deben ser colocadas en un servidor Web para que otros usuarios puedan acceder a ellas.

Desde los inicios hasta la actualidad el lenguaje HTML ha evolucionado pasando desde un lenguaje estático hasta un lenguaje más dinámico, modificaciones aportadas sobre todo por las personas que lo utilizan. Así, una evolución en el lenguaje suele surgir de una propuesta que es adoptada por algunos clientes

(browsers) y finalmente, con el uso, se verá si es eficiente y es adoptada y si es así, finalmente se incorpora al estándar.

Etiquetas básicas HTML.

Las etiquetas o tags son las encargadas de indicar a los exploradores Web cómo tienen que mostrar el texto y los gráficos. Estas etiquetas normalmente se escriben entre los símbolos < y > y suelen haber dos etiquetas una que indica el comienzo y otra que indica el fin (<>; </>).

Dentro del documento hay dos zonas bien diferenciadas: el encabezamiento, delimitado por **<HEAD>** y **</HEAD>**, que sirve para definir diversos atributos de la página y el cuerpo, delimitado por **<BODY>** y **</BODY>**, donde reside la información del documento.

2.9.2 PHP

Actualmente existen una gran variedad de tecnologías diseñadas para desarrollar aplicaciones web. Dentro de esta gran variedad hay herramientas que necesitan de una gran cantidad de líneas para generar una página web, así como otras que para la misma acción solo necesitan de una sola línea, PHP es una de esas herramientas debido a su sencillez, potencia, capacidad multiplataforma y economía tanto de adquisición (es gratuito) como de explotación (reducido consumo de recursos).

Es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. Su nombre oficial es PHP: HyperText Preprocessor¹², es la

¹² El nombre PHP tiene su origen a principios de 1995 en un conjunto de macros conocido como Personal Home Page Tools.

versión libre del sistema equivalente de Microsoft ASP, otras de sus ventajas son el alto rendimiento y facilidad de aprendizaje.

PHP es un lenguaje de programación que contiene muchos conceptos de C, Perl y Java. Su sintaxis es muy similar a la de estos lenguajes, haciendo muy sencillo su aprendizaje¹³.

Características fundamentales del lenguaje

PHP es un lenguaje para la creación de sitios web del que se pueden destacar las siguientes características:

- Es un potente y robusto lenguaje de programación embebido en documentos HTML, De forma que se pueden introducir instrucciones php dentro de las páginas. Gracias a esto el diseñador gráfico de la web puede trabajar de forma independiente al programador. PHP es interpretado por el servidor (apache) generando un HTML con el resultado de sustituir las secuencias de instrucciones PHP por su salida.
- Dispone de librerías de conexión con la gran mayoría de los sistemas de gestión de base de datos para el almacenamiento de información permanentemente en el servidor.
- Proporciona soporte a múltiples protocolos de comunicaciones en Internet (HTTP, IMAP, FTP, LDAP, SNMP, etc).
- Código fuente abierta: el código del intérprete está accesible para permitir posibles mejoras o sugerencias acerca de sus desarrollo (PHP ha sido escrito en lenguaje C).
- Portable y multiplataforma: existen versiones del intérprete para múltiples plataformas (Windows 95, 98, NT, 2000, Unix, Linux, etc.), esto permite que

¹³ GIL Javier y Otros. Creación de sitios Web con PHP 4.

las aplicaciones puedan ser portadas de una plataforma a otra sin necesidad de modificar ni una sola línea de código.

- PHP permite, una fácil interacción con un gran número de sistemas de bases de datos relacionales (Oracle, DB2, MySQL, PostgreSQL, etc.), mientras mantiene una sintaxis sencilla y clara.
- Dispone de facilidades para el procesamiento de ficheros, funciones de tratamiento de textos, generación dinámica de imágenes, tratamiento de documentos XML etc.

2.9.3 JAVASCRIPT

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento, gracias a su compatibilidad con la mayoría de los navegadores modernos, con Javascript se puede crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza.

Entre las acciones típicas que se pueden realizar en Javascript se tienen dos vertientes. Por un lado los **efectos especiales** sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar

instrucciones como respuesta a las acciones del usuario, con lo que podemos crear **páginas interactivas** con programas como calculadoras, agendas, o tablas de cálculo, etc.

2.9.4 JAVA

Java es un lenguaje de programación de alto nivel con el que se pueden escribir tanto programas convencionales como para Internet. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que el mismo programa escrito para Linux puede ser ejecutado en Windows sin ningún problema. Además es un lenguaje orientado a objetos que resuelve los problemas en la complejidad de los sistemas, entre otras.

La tecnología Java está compuesta básicamente por 2 elementos: el compilador y su plataforma. El compilador produce un código de bytes que se almacena en un fichero para ser ejecutado por el intérprete Java denominado máquina virtual de Java (Java Virtual Machine). Los códigos de bytes de Java son un conjunto de instrucciones correspondientes a un lenguaje máquina que no es específico de ningún procesador, sino de la máquina virtual de Java (Plataforma).

Sus objetivos principales están:

- Facilitar un entorno interpretado para:
 - 1) Aumentar la velocidad de desarrollo.
 - 2) Generar código portable.
- Permitir al usuario ejecutar más de una hebra a la vez.
- Soportar cambios dinámicos en tiempo de ejecución.

- Proporcionar más seguridad.

Applets

Desde el punto de vista de código Java se puede escribir como pequeños programas diseñados para ser ejecutados en el contexto de una página Web y, desde el punto de vista de una página Html representa una pequeña ventana gráfica dentro de la misma.

Por otra parte, la diferencia entre una aplicación Java y un Applet radica en cómo se ejecutan. Sabemos que para cargar y ejecutar una aplicación Java se utiliza directamente el intérprete Java, en cambio, un Applet se carga y ejecuta desde cualquier explorador que soporte Java.

Características del lenguaje

- Trabaja con programación orientada a objetos (POO).
- Fácil de aprender, partiendo de la base de C++.
- Es seguro y confiable, ya que posee las llamadas Excepciones, para la captura de un gran número de errores.
- Incluye instrucciones para trabajar procesos concurrentes, es decir procesos que se ejecutan compartiendo los recursos de la máquina.

2.10 BASES DE DATOS

Manejar grandes cantidades de datos e información en las organizaciones, ya sean de tipo administrativos, comerciales, contables, de inventarios, de clientes, etc, es inevitable, puesto que la información se encuentra presente en todas las actividades del hombre. Anteriormente, estos datos eran almacenados en grupos de registros en archivos separados donde se generaban problemas tales como la duplicación innecesaria de datos, la incompatibilidad entre archivos, entre otros. Por lo anterior, se vio la necesidad de organizar las estructuras de datos y crear un sistema de bases de datos que recopile la información de una manera mas coordinada y que, a su vez, genere confianza y seguridad en el manejo de la información.

Las bases de datos es una colección de datos relacionados entre sí, clasificados, estructurados y almacenados en uno o varios ficheros, para ser usados por los programas con el fin de facilitar su mantenimiento y acceso de una forma estándar” y un sistema de bases de datos “es una herramienta Software que presta servicios de soporte de almacenamiento de la base de datos. La información almacenada en la Base de Datos se organiza en campos (atributos) y registros (entidades), donde un campo se refiere a un tipo de información, y un registro a toda la información sobre un individuo.

Modelos de bases de datos

Los modelos de bases de datos son esquemas conceptuales encargados de representar y organizar los elementos de datos y sus relaciones. Existen varios modelos que son el Entidad-Relación, el modelo jerárquico de bases de datos, el modelo de bases de datos en red, entre otros.

Modelo de red

Es un modelo gráfico que representa la información por medio de nodos y se relacionan entre sí por medio de flechas

Modelo jerárquico.

Es el mas antiguo de los modelos y es una estructura de árbol, como por ejemplo el organigrama de una empresa.

Modelo de Entidad-Relación (E/R)

Es el más usado y representa la información por medio de tablas. Cada tabla corresponde a una entidad de aplicación y cada fila corresponde una instancia de esa entidad.

Conceptos fundamentales:

Entidad: es un objeto concreto que se puede distinguir de otros objetos.

Atributo: es una información acerca de una entidad o una relación y sirve para identificar y describir las entidades. Los atributos deben tener las siguientes características:

- no tomar varios valores en un mismo instante.
- No requieran de más información.
- Sea de interés para el sistema de bases de datos.

Existen dos clases de atributos: los identificadores y los descriptores; los identificadores son las mismas llaves que a su vez pueden ser: llave primaria y llave foránea.

Llave primaria: es la que distingue de forma unívoca una tabla de las demás.

Llave foránea: es el atributo que es la llave primaria de otra tabla.

Relación: es la asociación entre entidades.

DIAGRAMA E/R.



2.10.1 MYSQL

Es un sistema de administración relacional de base de datos de código fuente abierto, actualmente el más popular, y además, soporta diversos lenguajes de programación tales como: C, C++, Java, Perl, PHP, entre otros

Las diferentes características de MYSQL son:

- Almacena los datos en tablas separadas; esto permite que MYSQL sea más rápido y flexible. Estas tablas están conectadas por relaciones definidas que hacen posible combinar datos de distintas tablas.
- Por ser un software de fuente abierta, es más fácil de conseguirlo y utilizarlo sin la necesidad de licencias, siempre y cuando no sea con fines comerciales.
- Puede trabajar en distintas plataformas y distintos sistemas operativos.
- Es multiproceso, es decir puede usar varias CPU si éstas están disponibles.

3. DISEÑO Y DESARROLLO DEL SISTEMA

FASE INICIAL

3.1 CONCEPTO INICIAL

3.1.1 Recopilación de Información

Una vez decidida la elaboración de la herramienta como apoyo al tema de Representación del Conocimiento en Inteligencia Artificial, se iniciaron las actividades de búsqueda y recopilación de información inicial relacionada con los temas que incluiría el proyecto. Esta información, recopilada al principio del proyecto no fue la única utilizada, ya que a medida que se iba avanzando en el desarrollo, surgía la necesidad de hacer nuevas búsquedas de información, tanto para el contenido como para la realización de la herramienta. Toda esta información fue recopilada de diversas fuentes como libros, artículos, Internet y tesis de grado.

3.1.2 Análisis y Especificación de Requerimientos

El modelo de desarrollo seleccionado para la elaboración del proyecto, adopta el enfoque metodológico empleado en el desarrollo software basado en modelos de ciclos de vida. Dicha metodología combina el diseño expresado a través del Lenguaje Unificado de Modelado, UML, con un modelo de ciclo de vida de desarrollo software de Prototipado Evolutivo.

El modelo de ciclo de vida de Prototipado Evolutivo, se escogió después de haber realizado un análisis de los modelos de ciclos de vida más utilizados en el desarrollo de proyectos software, para lo que se tuvo en cuenta inicialmente las siguientes características del proyecto **SARAI**:

- Los requerimientos y especificaciones del proyecto solo estaban descritos globalmente al inicio de éste.
- Las especificaciones debían poder ajustarse a posibles modificaciones y adaptaciones que se hagan en el transcurso del desarrollo del proyecto.

Teniendo en cuenta estas características, los modelos no evolutivos fueron descartados, pues éstos exigen un conocimiento exacto de los requerimientos de lo que se desea construir desde el inicio y requieren de una fuerte planificación, lo cual los hace muy poco flexibles. Así mismo, se necesita un profundo conocimiento de las herramientas a utilizar.

Con el planteamiento del problema y la información preliminar se definieron los requerimientos relacionados con el contenido, el entorno y el lenguaje de programación que se utilizarían en la elaboración de la herramienta.

Los requerimientos en cuanto al contenido teórico establecidos serían las definiciones, sintaxis, reglas de formación e inferencia, de las cuatro formas básicas de representación del conocimiento escogidas para trabajar el proyecto: Redes Semánticas, Lógica de Predicados, Reglas de Producción y Frames; además de presentar ejercicios resueltos y ejercicios para resolver, que permitirán al estudiante reforzar y aplicar conocimientos adquiridos previamente.

Otro requerimiento establecido sería la posibilidad de permitir al profesor ingresar y eliminar documentos que puedan complementar los contenidos de la herramienta para que el estudiante pueda acceder a ellos en el momento que lo requiera.

Debido a que la herramienta sería implantada en el Aula Virtual, su desarrollo debería seguir los lineamientos de software y los requerimiento sugeridos por su administrador, tales como la utilización de PHP como medio de conexión para la

base de datos, la cual se haría en MYSQL , HTML para el desarrollo de las páginas, y algunos Applets de Java que se utilizarían para resolver ejercicios.

3.1.3 Entrenamiento previo

Corresponde a las actividades de familiarización y estudio de los diferentes lenguajes y herramientas de programación, iniciando con JAVA por su grado de complejidad y el poco conocimiento que se tenía sobre este, UML útil para el diseño de la herramienta, Fireworks y Dreamweaver necesarias para la elaboración y mejoramiento de páginas Web y finalizando con PHP y MYSQL, para permitir el ingreso, visualización y eliminación de documentos en la herramienta.

FASE ITERATIVA

3. 2 DISEÑO DE LA HERRAMIENTA

3.2.1 Diagramas de Casos de Uso

Aula Virtual

El Aula Virtual es un ambiente web, que proporciona un espacio para estudiantes y profesores, de modo que puedan interactuar de una forma diferente a la de una clase presencial.

Está compuesta por diferentes perfiles de usuarios, tales como:

- **Administrador:** usuario que se encarga del mantenimiento del sistema. Utiliza el sistema para crear, modificar y eliminar cursos, usuarios y períodos.

- **Estudiante:** usuario inscrito en uno o varios cursos del sistema, usa el sistema para desarrollar los proyectos, discutir acerca de estos, comunicarse con los demás usuarios y utilizar las diversas herramientas disponibles.
- **Profesor:** este usuario conduce la actividad académica del sistema. Utiliza el sistema para crear, modificar y borrar temas del contenido del curso, crear, modificar y borrar proyectos colaborativos y comunicarse activamente con los estudiantes.
- **Visitante:** es una persona externa al sistema, que desea obtener información de este. Utiliza el sistema para visualizar la información de los cursos ya terminados, sus proyectos y el desarrollo de los mismos.

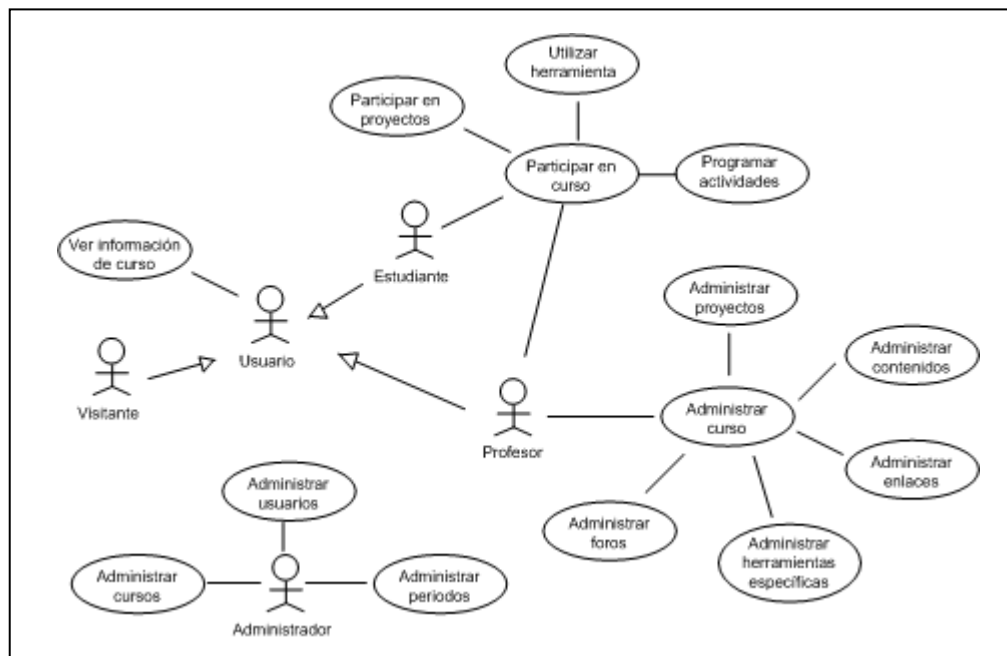


Figura 10. Diagrama de Casos de Uso generales del Aula Virtual

SARAI (Software de Apoyo al Aprendizaje en Línea de Formas Básicas de Representación del Conocimiento en Inteligencia Artificial)

Como primera instancia se identificarán los actores que intervienen en el sistema



Figura 11. Actores que intervienen en SARAI

- **Estudiante**

Usuario inscrito a uno o varios cursos del sistema, donde esté habilitada la herramienta. Hace uso de esta para ver contenidos, realizar ejercicios y abrir documentos.

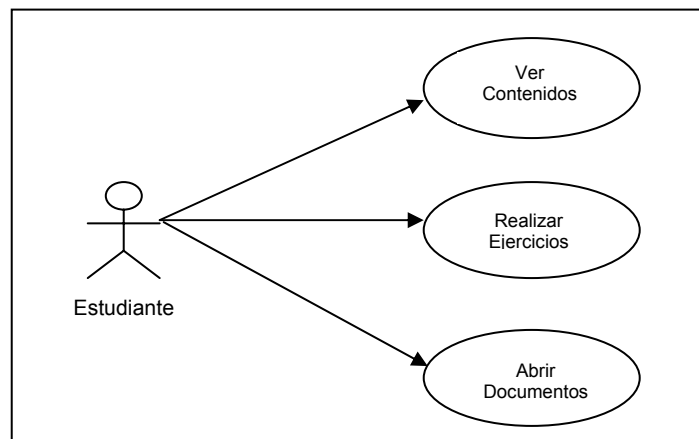


Figura 12. Diagrama de Casos de Uso – Estudiante

- **Profesor**

Este usuario utiliza la herramienta para ver contenidos, realizar ejercicios, ingresar, borrar y abrir documentos.

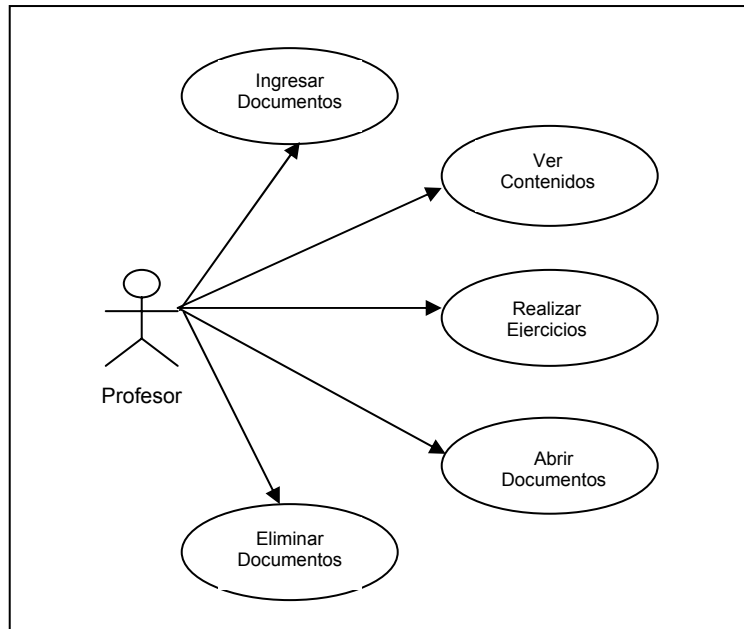


Figura 13. Diagrama de Casos de Uso - Profesor

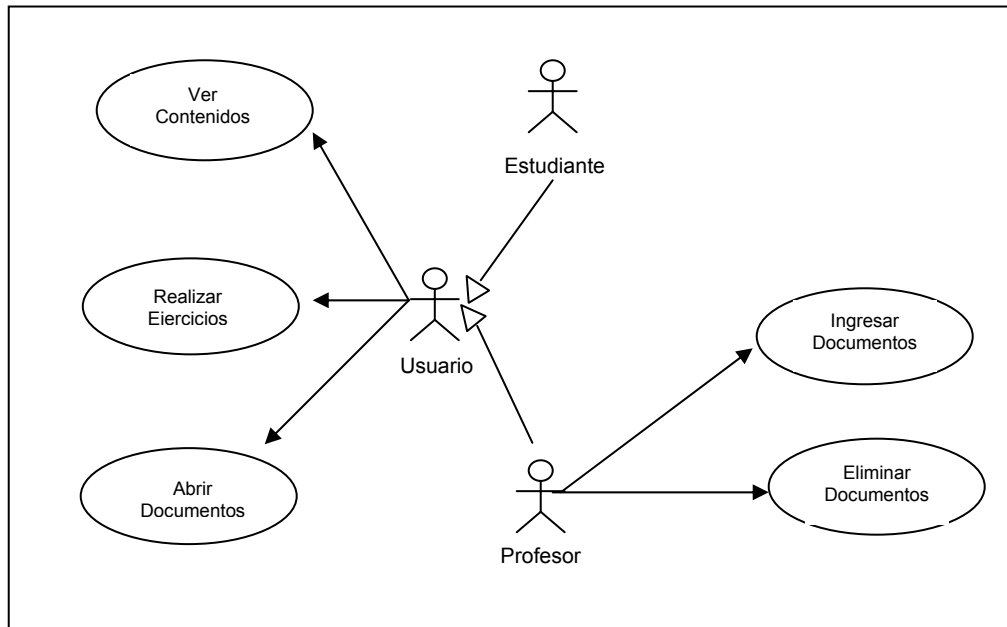


Figura 14. Diagrama Casos de Uso General SARAI

3.2.2 Diseño de Interfaz

La interfaz es la zona de comunicación entre el usuario y la máquina, de allí su importancia para la educación. Si los mensajes no los comprende el usuario o no se adaptan a sus características especiales, el software no generará interés o motivación en los estudiantes. Los menús, textos, gráficos, balance de colores, contribuyen a mejorar o a obstaculizar la interacción usuario-máquina.

Para la elaboración de SARAI se hicieron tres interfaces gráficas; La primera interfaz el usuario encuentra el respectivo logo y nombre de la herramienta, logotipos de las entidades interesadas y responsables de su desarrollo, así como un enlace que invita al usuario a explorar la herramienta o salir de ella. Una vez se ingrese como tal a la herramienta se encontrará la segunda interfaz, correspondiente a dos menús, uno de opciones generales, otro de opciones específicas y sus respectivos contenidos; la tercera y última interfaz de visualiza desde un enlace del menú de opciones específicas de la segunda interfaz, el cual cuenta con un menú de opciones específicas y según el tipo de usuario que ingrese a la herramienta, esta interfaz mostrará las opciones de ingreso, eliminar y mostrar documentos, además de sus respectivos contenidos.

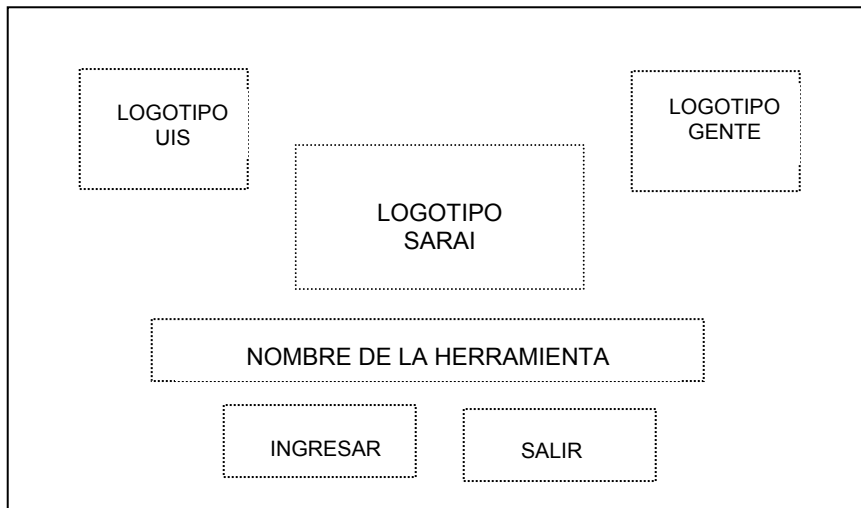


Figura 15. Interfaz Gráfica Inicial

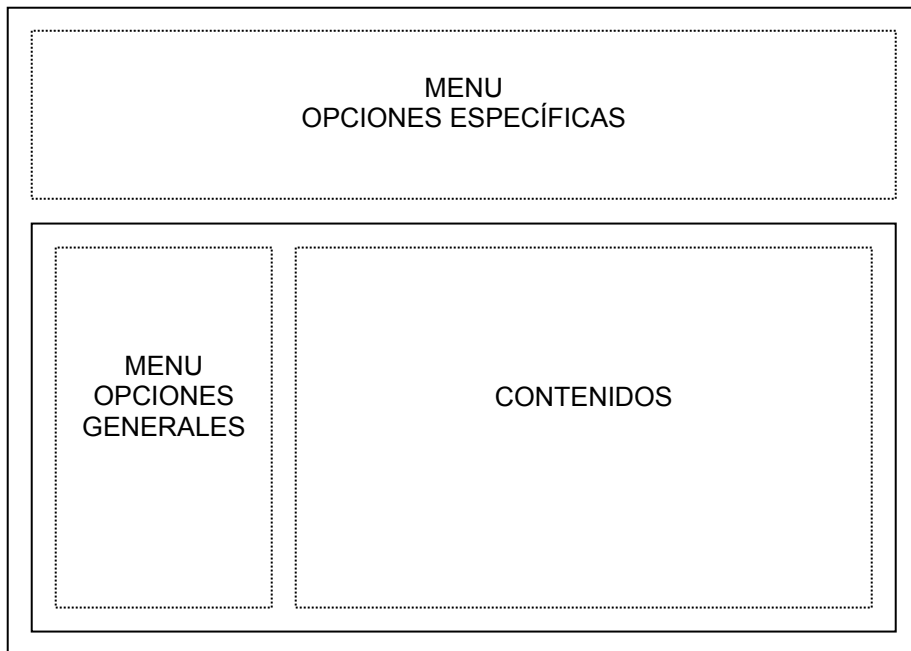


Figura 16. Interfaz Gráfica General

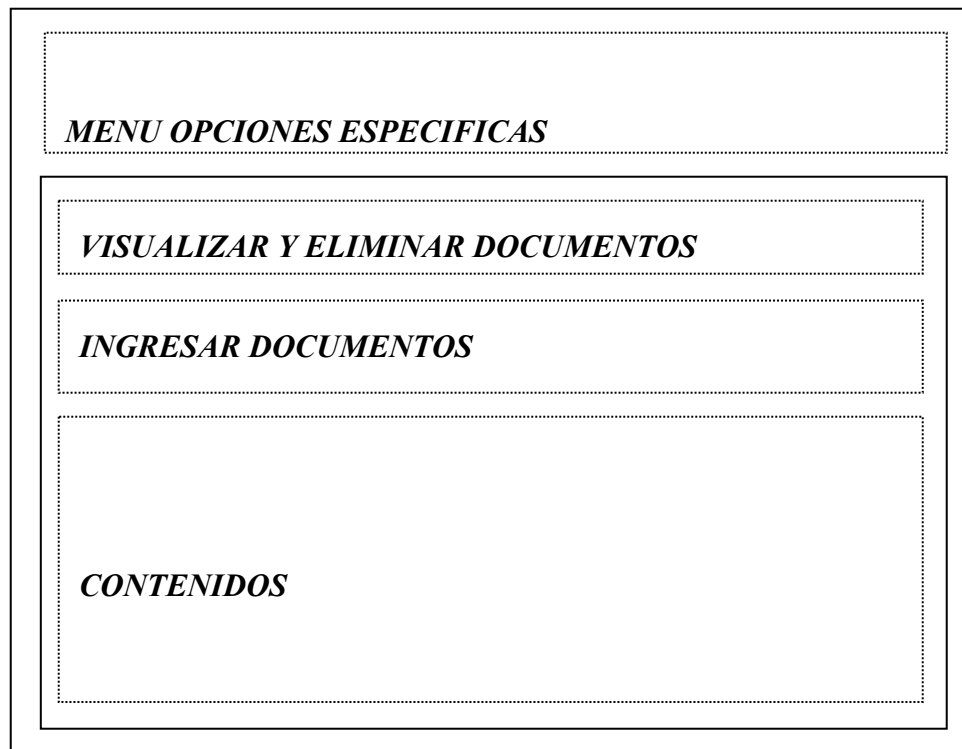


Figura 17. Tercer Interfaz Gráfica – Profesor

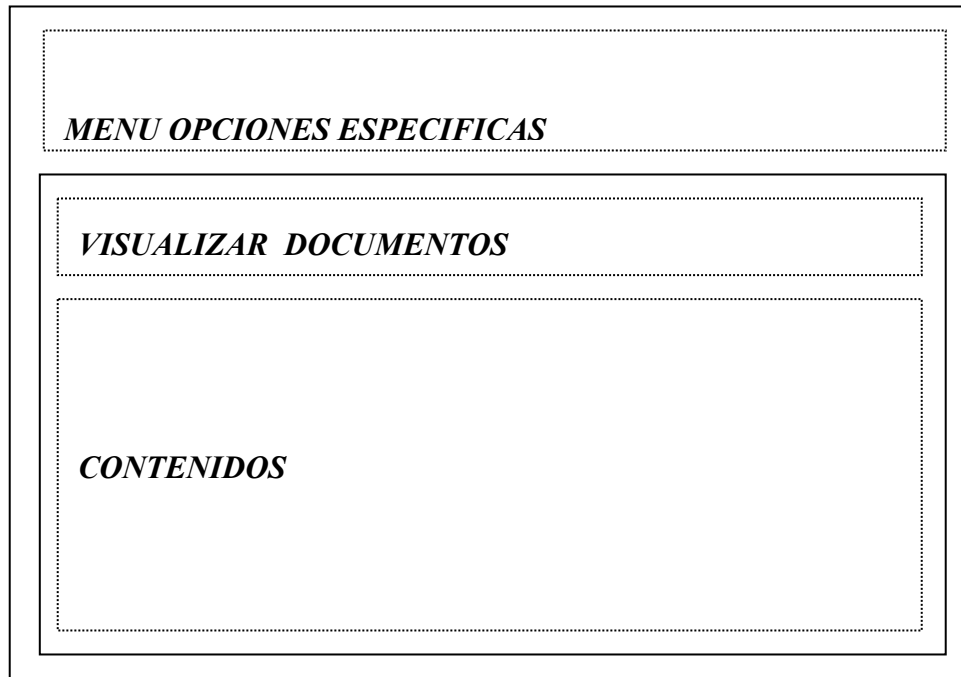


Figura 18. Tercer Interfaz Gráfica – Estudiante

3.2.3 Diseño de la Base de Datos

Para permitir la constante interacción entre el alumno y el estudiante y la actualización de la herramienta fue necesario la construcción de una pequeña base de datos hecha en MYSQL, llamada *Ejemplo*, que permite al profesor ingresar, visualizar y eliminar documentos. Como el único propósito de la base de datos es guardar las direcciones y nombres de los documentos, no hubo necesidad de realizar integridad referencial sobre sus tablas.

La base de datos dispone de las siguientes tablas:

- Ejereslogica
- Ejeresframes
- Ejeresreglas
- Ejeresredes

Las cuatro tablas constan de los mismos nombre de campos:

NOMBRE DEL CAMPO	TIPO DE CAMPO	DESCRIPCION
CODIGO	<i>Integer</i> Auto_Increment	Corresponde a la llave primaria de la tabla
ENLACE	Varchar (50)	Corresponde a la dirección y nombre del documento
COMENTARIO	Varchar (100)	Breve descripción del contenido del documento

Tabla 2. Campos de las tablas de la Base de Datos

3.3 IMPLEMENTACION DEL PRIMER PROTOTIPO

Una vez terminado el diseño, y con los requerimientos establecidos, se procedió a crear las páginas HTML, con los diseños ya determinados, la creación de la base de datos y sus respectivas instrucciones en PHP, inmersas en el código HTML.

3.3.1 Código Fuente PHP para Conexión con la Base de Datos

```
<?php
$conex = mysql_connect("localhost","root", "")
or die("no se pudo realizar la conexion");
mysql_select_db("ejemplo", $conex);
$fecha = date("Y_m_d_hi_");
?>
```

3.3.2 Código Fuente PHP para Cerrar la Conexión con la Base de Datos

```
<?php
mysql_free_result($resultado);
mysql_close($conex);
?>
```

3.3.3 Código Fuente que Permite el Ingreso de Nuevos Documentos

Este código es el mismo para las cuatro tablas y solo es visible para el profesor.

```
<?php
if ($accion == 'ingresar')
{
if (!(empty($enlace)) && !(empty($comentario)))
{
    $nombre_archivo =HTTP_POST_FILES['enlace']['name'];
    $nombre_archivo=$fecha .      ereg_replace      ("^[a-zA-Z0-
9_\.]", "", $nombre_archivo);

    $cons2= "insert into ejeresframes (enlace, comentario) values
('$nombre_archivo','$comentario');";

    $resultado = mysql_query($cons2,$conex);
    if ($resultado)
    echo '<br><font color=#333399><h4><strong>LOS CAMPOS HAN SIDO BIEN
INSERTADOS</h4></font ></strong></br>';
    }
    else
    echo '<font color=#333399><h4><strong>DEBE LLENAR TODOS LOS
CAMPOS</h4></font></strong>';

?>
```

3.3.4 Código Fuente que Permite Eliminar Documentos

Este código es el mismo para las cuatro tablas y solo es visible para el profesor.

```
<?php
if ($accion == 'borrar')
if ($eje=='')
{ echo "<br><font color=#333399><h4><strong>NO HAY CONTENIDOS QUE
ELIMINAR</h4></font ></strong></br>";}
else {
    $new = array_values($eje);
    $num = count($new);
    for ( $i= 0 ; $i < $num ; $i++) {
    $borrar=      "delete      from      ejeresframes      where      codigo
=".$new[$i].".";";
        $resultado1 = mysql_query($borrar,$conex);
    }}}}
?>
```

3.3.5 Código Fuente que Permite Visualizar Documentos

Este código es el mismo para las cuatro tablas y solo es visible tanto para el profesor, como para el estudiante.

```
<?php
echo "<TABLE BORDER=1 WIDTH=90%> <TR>
<TD
ALIGN='CENTER'><FONTCOLOR='#333399' size='4'><strong>Enlace</FONT><
/strong></TD>
<TD
ALIGN='CENTER'><FONTCOLOR='#333399' size='4'><strong>Comentario</FO
NT></strong></TD>
<TD
ALIGN='CENTER'><FONTCOLOR='#333399' size='4'><strong>*</FONT></stro
ng></TD>
</TR>";
$cons = "select * from ejeresframes";
$resultado = mysql_query($cons,$conex);
$num_campos = mysql_num_fields($resultado);
$num_filas = mysql_num_rows($resultado);

for ($j = 0 ; $j < $num_filas ; $j++){ echo "<TR>";
echo "<TD ALIGN='center'>";
$atributo=mysql_result($resultado, $j, "enlace");
echo "<a href=\"ejercicios_sarai/$atributo\".\"\"target=\"_blank \"
>$atributo</a>\n";
echo "</td>";
echo "<TD ALIGN='center'>";
echo mysql_result($resultado, $j, "comentario");
echo "</td>";
echo "<TD ALIGN='center'>";
echo " <input name=\"eje[\".$j.\"}\" type=\"checkbox\" value=\"\".
mysql_result($resultado, $j, "codigo").\"\">";
echo "</td>"; echo "</tr>";
echo "\n"; }
if ($num_filas == 0){
echo "<td colspan=3 ><center>";
echo "<br><font color=#333399><strong>NO HAY CONTENIDOS
DISPONIBLES</font ></strong></br>";
echo "</center></td></tr>\n";
}
echo "</table>\n";
echo "</center>";
?>
```

3.3.6 Código Fuente para Validar el Tipo de Usuario

En el Aula Virtual se maneja la variable `SESSION_TYPE`, que valida el tipo de usuario que ingresa, el cual si tiene el valor de dos, es porque el usuario es profesor, de lo contrario es un estudiante.

```
<?php
    session_start();
    if ($SESSION_TYPE!=2){
        header("location: EjeFAl.php");
        exit();
    }
?>
```

3.4 PRUEBAS DEL PRIMER PROTOTIPO

Habiendo terminado el primer prototipo, se procedió a montarlo en el Aula Virtual, para realizar las pruebas correspondientes, sobre el funcionamiento de las páginas Web, la base de datos, el manejo de las sesiones y los Applets. Observando un buen desempeño de la herramienta en cuanto a su funcionalidad, pero en la página de lógica de predicados no se pudieron observar algunos contenidos.

3.4.1 Entrega del primer prototipo

Con la realización de las pruebas correspondientes y observando que casi la totalidad del software funcionaba satisfactoriamente acorde a los requerimientos preestablecidos; se procedió a hacer una revisión en compañía de la directora del proyecto quien es la encargada y capacitada para dar su punto de vista respecto al diseño general de la herramienta y evaluar el buen desempeño, en cuanto al manejo de la sesión del profesor, sin encontrar problema alguno, respecto a la administración de documentos; finalmente sugirió, algunos cambios en los Applets

y dio pautas para el desarrollo del logotipo de SARAI que hasta ese momento no se tenía.

3.5 DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO FINAL

En cuanto al diseño de la herramienta, no se realizaron modificaciones a los diagramas de casos de uso, diseño de interfaces y bases de datos, a los establecidos inicialmente.

Para su implementación, una vez detectados los errores de funcionamiento del primer prototipo, se iniciaron los respectivos arreglos para dar paso al prototipo final. Se encontró que el error que aparecía en la página de Lógica de Predicados, al no visualizar algunos contenidos, era producto de fallas en el montaje de SARAI en el Aula Virtual, y no de la herramienta como tal. Además, se hicieron los respectivos cambios sugeridos por la directora del proyecto en cuanto a los Applets y la interfaz gráfica. Para dar como resultado el prototipo final.

FASE FINAL

3.6 IMPLANTACIÓN DEL PROTOTIPO FINAL.

SARAI está implantada en el Aula Virtual como una herramienta específica, de la cual se despliega una ventana, donde se encuentran otras aplicaciones.

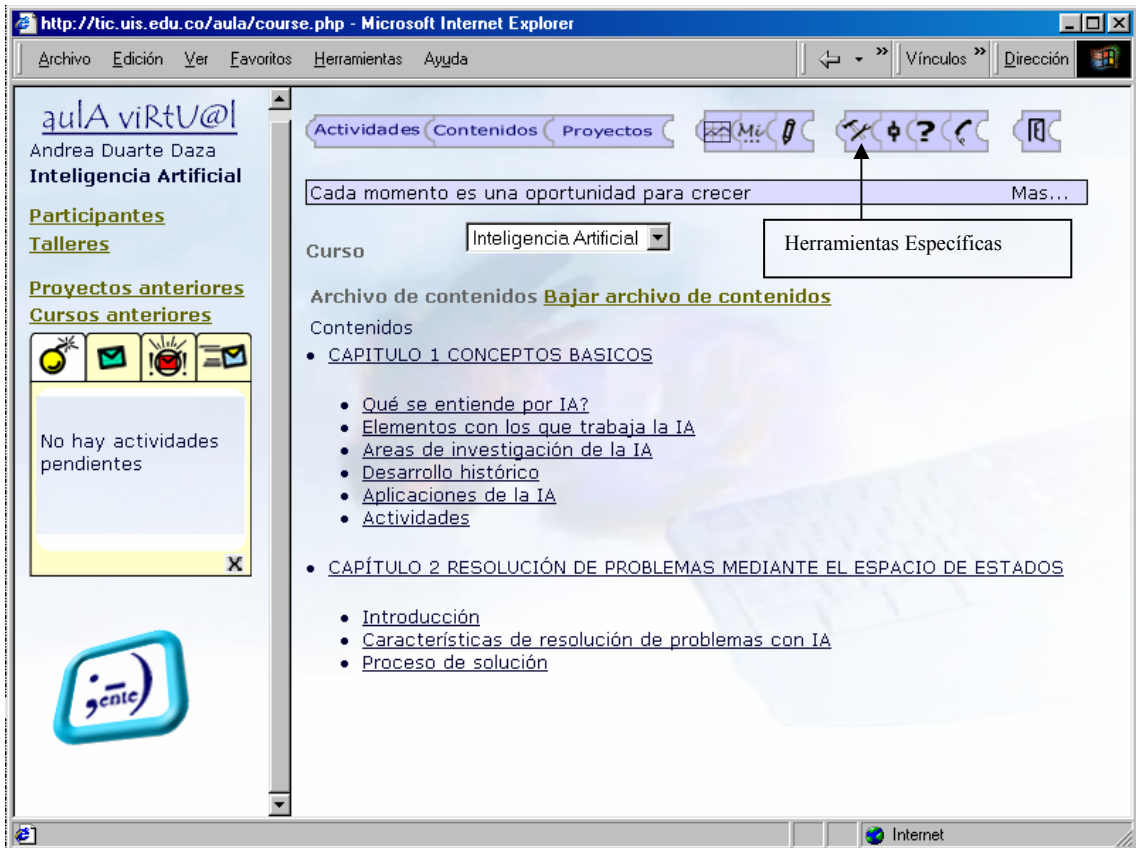


Figura 19. Vínculo a la ventana Herramientas

3.6.1 Código Fuente para Acceder a la Herramienta

Applet

```
import java.awt.*;
import java.applet.*;
import java.net.*;

public class Sarai extends Applet {
```

```

String directorio;

public void init() {
    directorio = getParameter("PATH");
    try {
        URL URLSarai = new URL(directorio + "/" +
"index.htm");
        getAppletContext().showDocument(URLSarai);
    } catch (MalformedURLException mue) {
    } } }

```

Página HTML

```

<HTML><BODY>
<APPLET CODE = Sarai.class width="500" height="250">
<PARAM NAME = PATH VALUE =
"http://tic.uis.edu.co/aula/tools/sarai">
</APPLET>
</BODY></HTML>

```



Figura 20 Ubicación de SARAI en el Aula Virtual

3.6.2 Requisitos para el uso de la herramienta

Para lograr los objetivos de SARAI, es necesario que el usuario que lo vaya a utilizar, posea conocimientos previos sobre Inteligencia Artificial, Representación del Conocimiento y sus formas básicas: Redes Semánticas, Lógica de Predicados, Reglas de Producción y Frames. Tener conocimientos básicos sobre el manejo adecuado del computador.

Para la visualización de los Applets es indispensable que el usuario disponga de la máquina virtual de Java en su computador.

4. DESCRIPCIÓN DE LA HERRAMIENTA

4.1 PANTALLA DE INICIO



Figura 21. Pantalla de Inicio

Al cargar la herramienta se presenta la pantalla de inicio, que consta de la información de las entidades vinculadas con el software, su nombre y logotipo, dos botones de ingreso y salida de la misma.

4.2 PANTALLA PRINCIPAL



Figura 22. Pantalla Principal

Esta pantalla principal es la que hace una descripción de la Representación del Conocimiento y consta de tres frames;

4.2.1 Frame Superior



Figura 23. Frame Superior de Pantalla Principal

Está compuesta por los logotipos: SARAI, UIS, un link que permite regresar a la pantalla inicial de la herramienta, además del título de Representación del Conocimiento.

4.2.2 Frame Central

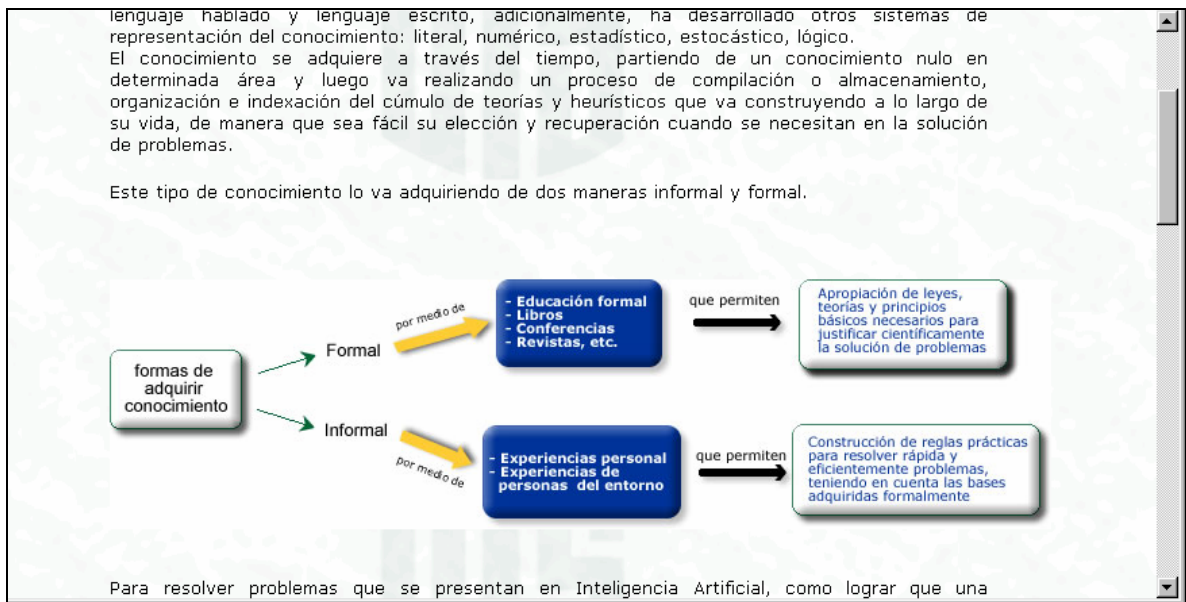


Figura 24. Frame Central de Pantalla Principal

Contiene toda la información acerca de Representación del Conocimiento, definición, características etc.

4.2.3 Frame Lateral



Figura 25. Frame Lateral de Pantalla Principal.

Contiene el menú general, con cuatro links que conducen hacia las cuatro formas de representación del conocimiento, este menú permanece estático para todas las demás páginas y un link final que permite la salida de la herramienta.

4.3 PANTALLAS DE LAS FORMAS DE REPRESENTACIÓN DEL CONOCIMIENTO

4.3.1 Pantalla Redes Semánticas

REDES SEMÁNTICAS

FRAMES

REGLAS DE PRODUCCIÓN

LÓGICA DE PREDICADOS

Redes Semánticas

EJERCICIOS VOLVER

DEFINICIÓN

Las redes semánticas, fueron originalmente desarrolladas para representar el significado o semántica de oraciones en inglés, en términos de objetos y relaciones.

Actualmente, el término redes asociativas es más ampliamente utilizado para describirlas ya que no sólo se las usa para representar relaciones semánticas, sino también para representar asociaciones físicas o causales entre varios conceptos u objetos. Son muy apropiadas para representar conocimiento de naturaleza jerárquica. Su concepción se basa en la asociación de conocimientos que realiza la memoria humana. Las principales aplicaciones son: comprensión de lenguaje natural, bases de datos deductivas, visión por computadora, sistemas de aprendizaje.

Las redes semánticas se caracterizan por representar el conocimiento en forma gráfica, su principal idea es que el significado de un concepto depende del modo en que se encuentre conectado con otros conceptos. En su construcción se destacan los siguientes componentes:

NODOS: se denominan también objetos (elementos del conocimiento). Pueden ser:

Objetos físicos:	se refieren a todo objeto que pueda verse y tocarse.
Objetos conceptuales:	hace alusión a hechos, eventos, cualidades.
Objetos descriptivos:	permiten asociar los nodos con los enlaces.

red Tiene Componen

Figura 26. Pantalla Redes Semánticas

Al igual que la pantalla principal la pantalla de Redes Semánticas consta de los mismos frames, sin embargo el frame superior varía además del título, en que contiene un link adicional que conduce hacia una nueva página de ejercicios, la cual se mostrará dependiendo del tipo de usuario.

Pantalla Ejercicios Redes Semánticas Estudiante



Redes Semánticas

CONTENIDOS DISPONIBLES

Enlace	Comentario
NO HAY CONTENIDOS DISPONIBLES	

EJERCICIOS RESUELTOS

Convertir en una única red semántica los siguientes enunciados:

- El avestruz es un ave que tiene patas largas y no puede volar.
- El águila es un ave que vuela muy bien.
- El ave es una animal que pone huevos, tiene plumas, vuela bien y puede respirar.

Figura 27. Pantalla Ejercicios Redes Semánticas Estudiante

Comprende de una cabecera principal con los respectivos logotipos, título de la página, tres links: ejercicio (para resolver; applet), Volver (Pantalla Redes Semánticas) y Salir de la herramienta. A continuación se muestra una tabla con los documentos disponibles para bajar, seguido de una serie de ejercicios resueltos, que permanecen fijos, con el fin de complementar y aplicar la teoría expuesta en la página anterior.

Pantalla Ejercicios Redes Semánticas Profesor

Redes Semánticas

CONTENIDOS DISPONIBLES

Enlace	Comentario	*
NO HAY CONTENIDOS DISPONIBLES		

INGRESO DE NUEVOS CONTENIDOS

Enlace: Examinar...

Comentario:

GUARDAR BORRAR

Figura 28. Pantalla Ejercicios Redes Semánticas Profesor

Su contenido es similar al de la página del estudiante, sin embargo, en la tabla con los documentos disponibles, se da la opción de poder eliminar los que se deseen, seguido de una segunda tabla que permite el ingreso de nuevos documentos al sistema.

Pantalla Ejercicio a Resolver Redes Semánticas

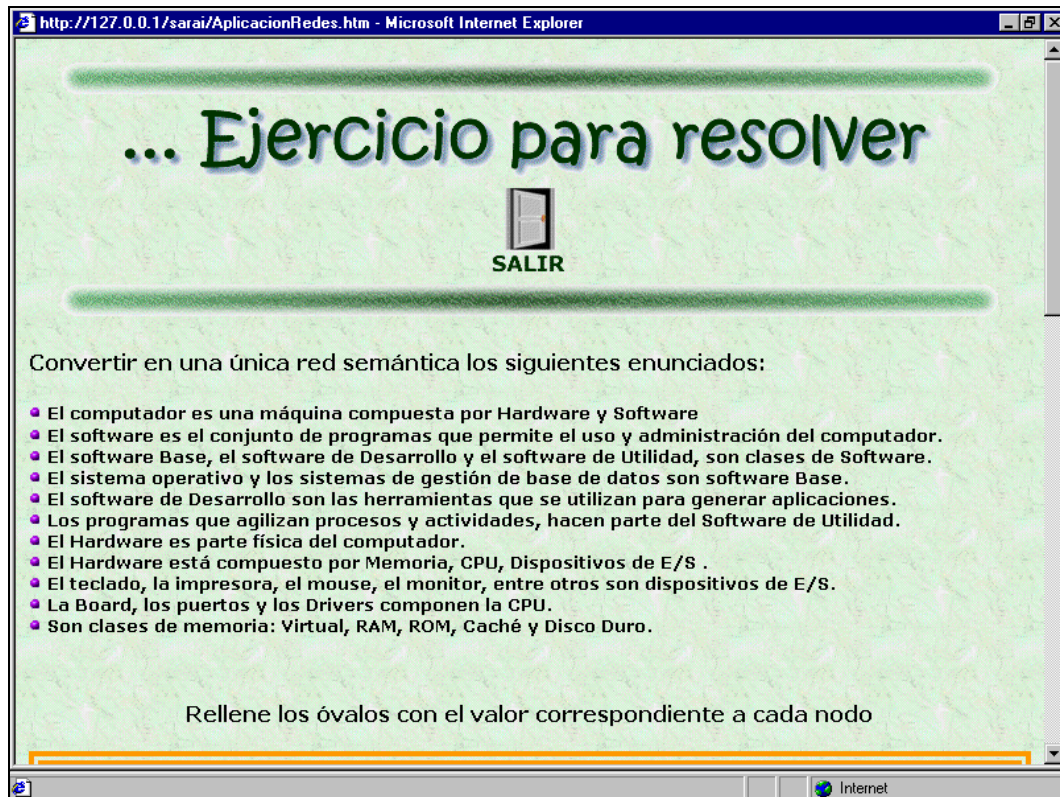


Figura 29. Pantalla Ejercicio a Resolver Redes Semánticas

Esta página contiene el enunciado de un ejercicio característico de esta forma de Representación del Conocimiento, para ser resuelto por el usuario de la siguiente forma:

Hay un applet que muestra una red semántica vacía, con sus nodos respectivamente numerados y para cada nodo hay una lista con opciones para escoger la respuesta que el usuario considere la adecuada para ese nodo.

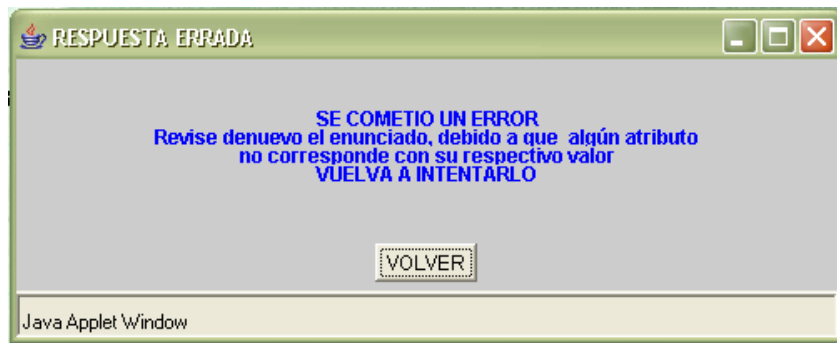


Figura 31. Ventana de Respuesta Errada

4.3.2 Pantalla Reglas de Producción

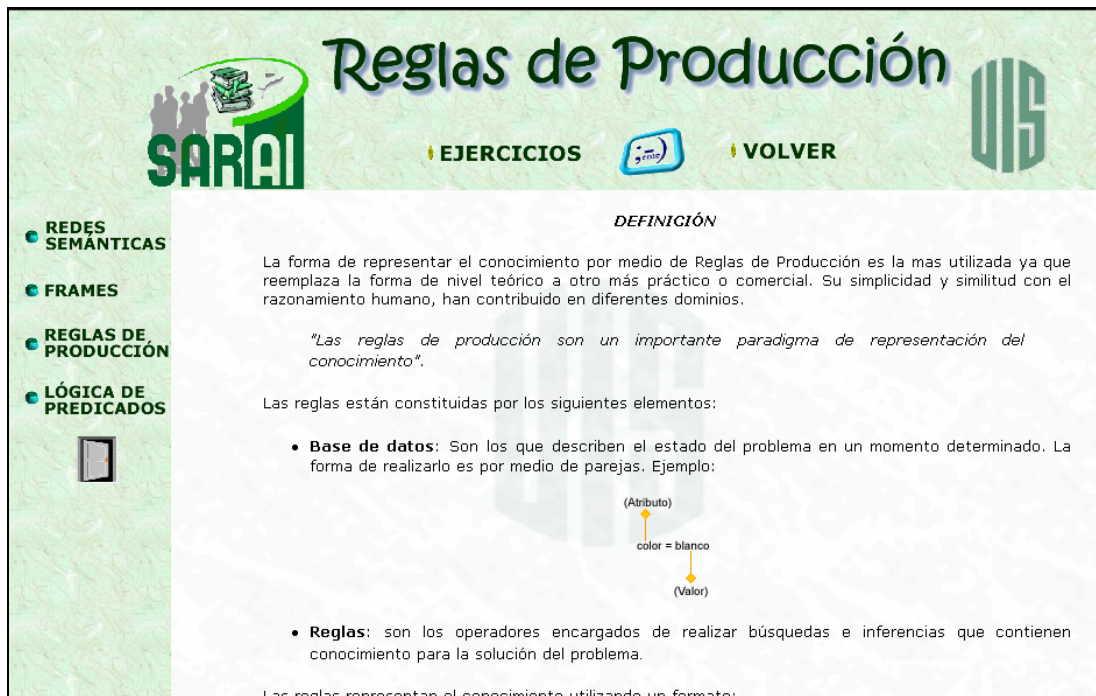


Figura 32. Pantalla Reglas de Producción

Esta pantalla es similar a la de todas las pantallas de las formas de Representación del Conocimiento y el link de ejercicios abre la página de ejercicios que tiene la misma estructura de todas, dependiendo del tipo de usuario.

Pantalla Ejercicio a Resolver de Reglas de Producción



Figura 33. Pantalla Ejercicio a Reglas de Producción

En esta pantalla se ofrece al usuario el enunciado de un ejercicio para representarlo por medio de Reglas de Producción escogiendo de una posible lista de respuestas, la que se considere correcta. En esta misma página se muestra un link que conduce hacia otro ejercicio para resolver por medio de encadenamiento hacia delante.

R12 SI	<input type="text"/>
ENTONCES	<input type="text"/>
R13 SI	<input type="text"/>
ENTONCES	<input type="text"/>
R14 SI	<input type="text"/>
ENTONCES	<input type="text"/>
R15 SI	<input type="text"/>
ENTONCES	<input type="text"/>
R16 SI	<input type="text"/>
ENTONCES	<input type="text"/>
R17 SI	<input type="text"/>
ENTONCES	<input type="text"/>
R18 SI	<input type="text"/>
ENTONCES	<input type="text"/>

Figura 34. Primer Ejercicio a Resolver Reglas de Producción

Segundo Ejercicio

Resolver el siguiente ejercicio utilizando encadenamiento hacia adelante. Teniendo en cuenta la estrategia de control de números con prioridad descendente, por ejemplo: $R7 > R3$, y Base de Datos (B.D) A,B con meta X.

R1: si A y C entonces F
R2: si E y C entonces G
R3: si H y F entonces E
R4: si E y F entonces D
R5: si G y H entonces I
R6: si D entonces G
R7: si A entonces C
R8: si B y F entonces H
R9: si I entonces X

Siguiendo el esquema:

B.D	A,B	Conclusión	→	→ ...	→ X
REGLAS	Regla				

B.D	A,B	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	X
REGLAS		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

Figura 35. Segundo Ejercicio a Resolver Reglas de Producción

4.3.3 Pantalla Frames



Figura 36. Pantalla Frames

Pantalla Ejercicio a Resolver de Frames

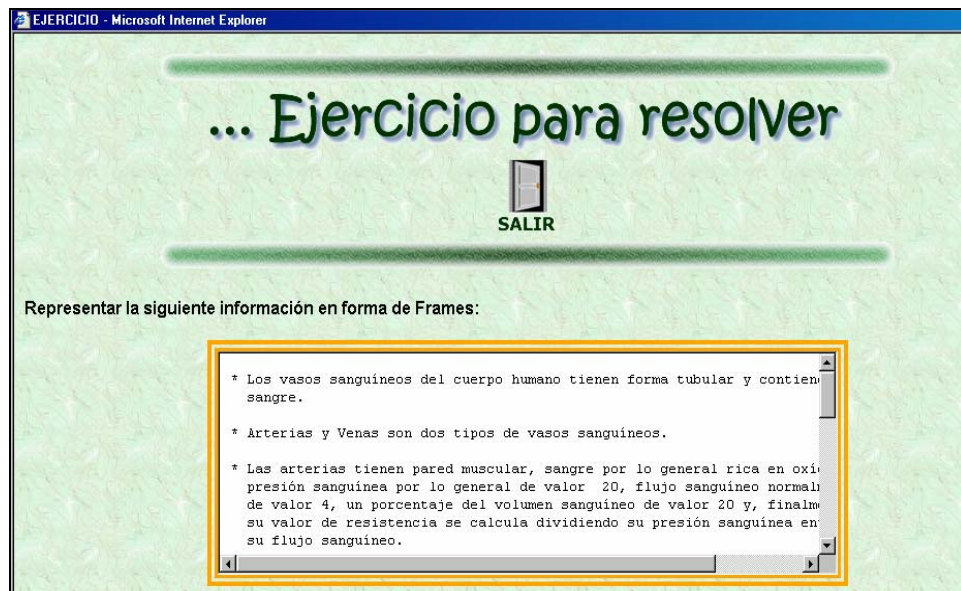


Figura 37. Pantalla Enunciado Ejercicio Frames

Se presenta al usuario un ejercicio, con su respectivo enunciado para ser resuelto escogiendo de una serie de botones, la respuesta que se considere correcta, al seleccionar el botón de validar, este deja o no continuar con el siguiente applet así sucesivamente hasta terminar el ejercicio, cuando este finaliza aparece un applet con el resumen de todo el ejercicio.

Figura 38. Ejercicio a Resolver Frames

4.3.4 Pantalla Lógica de Predicados

Figura 39. Pantalla Lógica de Predicados

Pantalla Ejercicio a Resolver de Lógica de Predicados

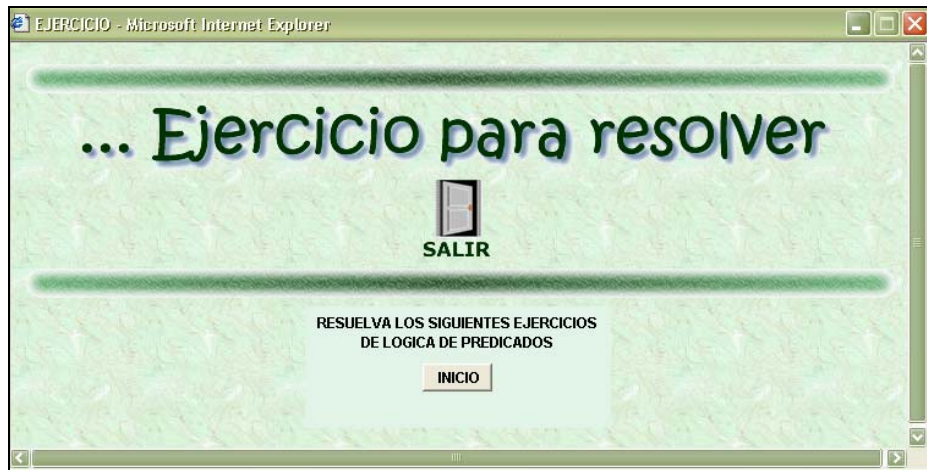


Figura 40. Pantalla Ejercicio a Resolver Lógica de Predicados

Se presenta al usuario tres ejercicios que saldrán de una forma aleatoria, dos de ellos consisten en tratar de convertir una oración a fbs de lógica y otro es tratar de resolver una frase por medio de Resolución.

Primer Ejercicio

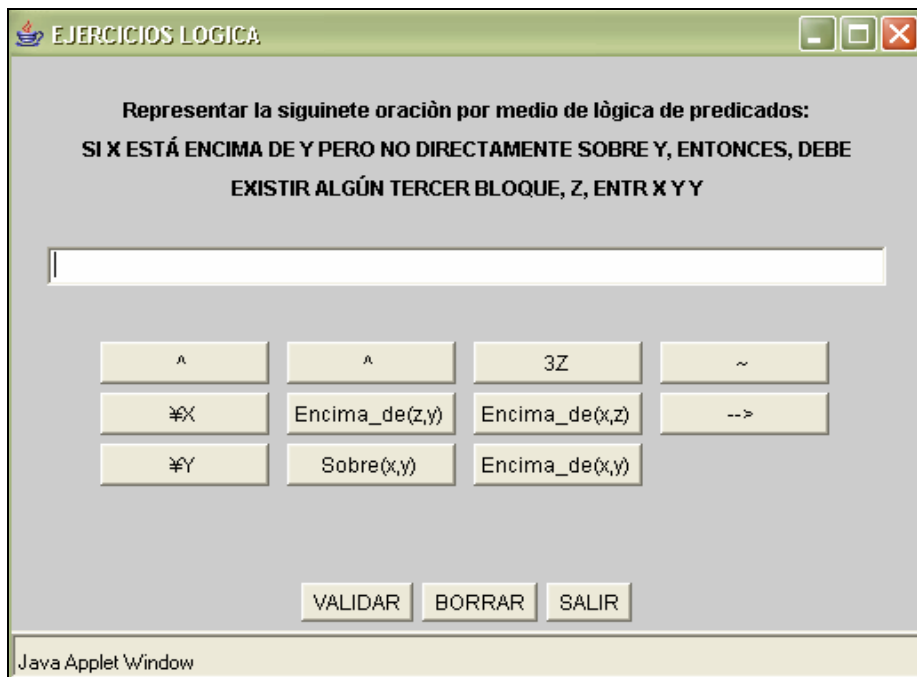


Figura 41. Primer Ejercicio Lógica

Segundo Ejercicio

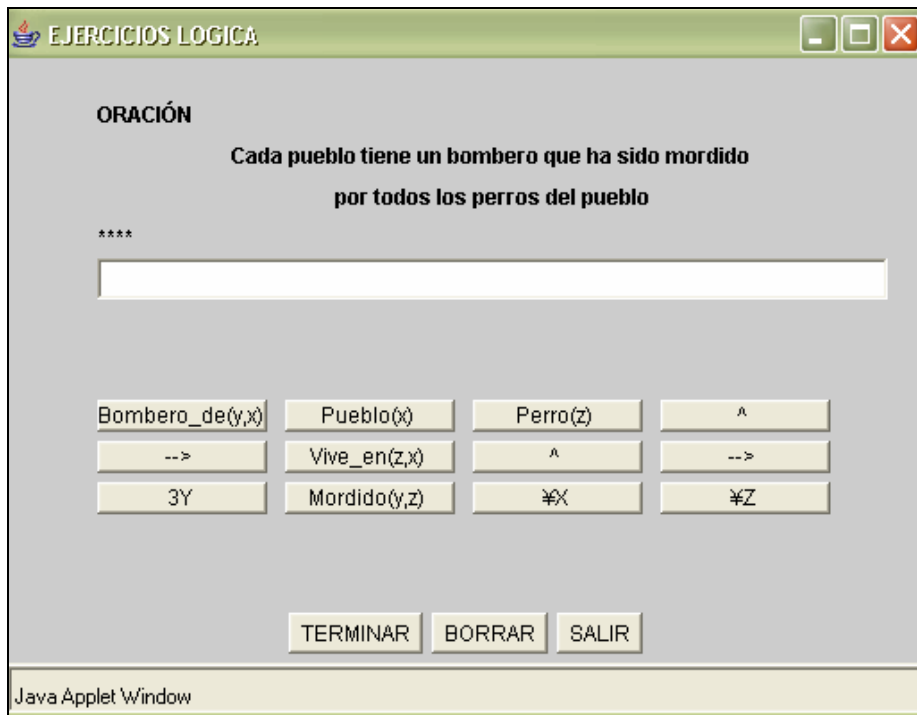


Figura 42. Segundo Ejercicio Lógica

Tanto el primer ejercicio como el segundo consiste en:

Dada una oración, el usuario deberá representarla por medio de lógica de Predicados hasta el momento que considere que esté terminada. El botón validar o terminar informará si el ejercicio está correcto. El botón borrar da la opción de volver a empezar y por último, el botón salir abandona el ejercicio.

Tercer Ejercicio

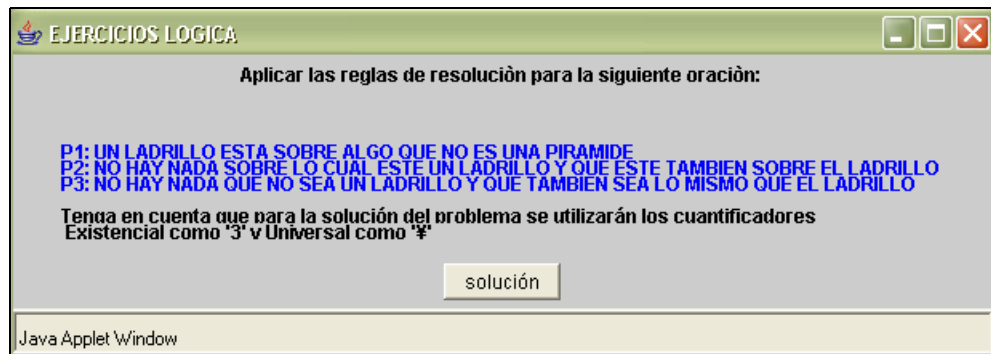


Figura 43. Tercer Ejercicio Lógica Pantalla inicial

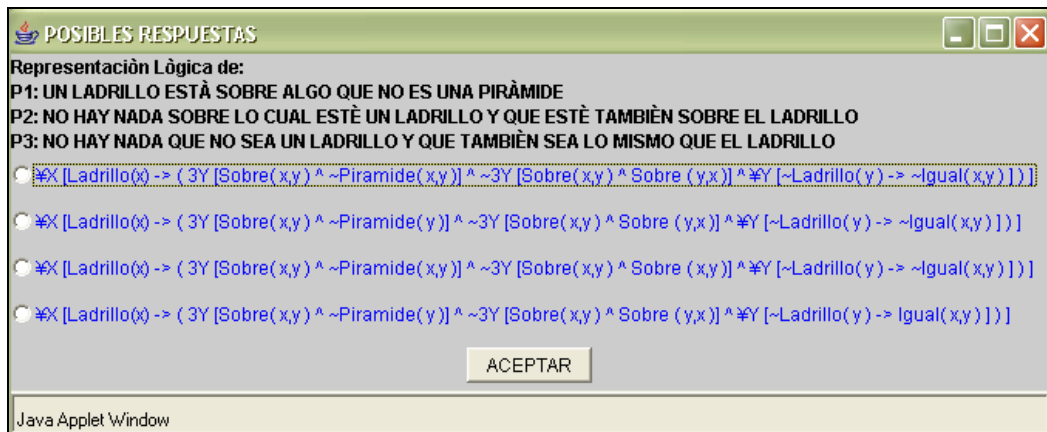


Figura 44. Tercer Ejercicio Lógica

Este Ejercicio busca que el usuario por medio de una serie de opciones aplique las reglas de resolución, donde al seleccionar el botón aceptar se valida si corresponde a la respuesta correcta.

5. CONCLUSIONES

- ❖ El uso de UML facilitó el desarrollo de la herramienta ya que a través de él, se comprendieron y expresaron de mejor forma las necesidades del cliente, identificando los roles de cada tipo de usuario, lo que aumentó la calidad del diseño del software.
- ❖ El Prototipado Evolutivo, que fue la metodología escogida al inicio del proyecto, fue la adecuada ya que los requerimientos y especificaciones del proyecto solo estaban descritos globalmente al inicio de éste y, además, las especificaciones se pudieron ajustar a las modificaciones y adaptaciones que se hicieron en el transcurso del desarrollo de la herramienta.
- ❖ El uso de los Applets de Java y PHP permitió la creación de páginas web dinámicas, lo que facilita que el usuario se interese mas por explorar la herramienta para conseguir el afianzamiento sobre el tema de Representación del Conocimiento.
- ❖ El uso de software de libre distribución constituye una buena alternativa en la realización de herramientas a bajo costo, por su fácil adquisición, su calidad y su gran funcionalidad.
- ❖ El desarrollo de herramientas Web que apoyan el proceso de aprendizaje en los estudiantes, y que permiten la constante interacción alumno-profesor, se han convertido en una necesidad actual que debe ser muy bien aprovechada, pues el avance tecnológico unido con la cultura informática así lo requiere.

- ❖ La educación virtual es una alternativa en el proceso de enseñanza-aprendizaje, muy importante en la actualidad ya que aprovecha el potencial que ofrecen las Tecnologías de Información y Comunicación .

6. RECOMENDACIONES

- ❖ Continuar con la creación de nuevas herramientas, como SARAI, que apoyen el proceso de aprendizaje de una forma virtual en asignaturas, tanto de la carrera de Ingeniería de Sistemas como en otras, para enriquecer los procesos de formación.
- ❖ Es necesario utilizar la herramienta en las experiencias del aula de clases, donde el alumno y el profesor, entregarán aportes de acuerdo con su experiencia y que permitan definir requerimientos y correcciones que posibiliten mejorar las especificaciones del software.
- ❖ Para próximas versiones de SARAI se recomienda:
 - a) Incluir una aplicación que pueda convertir información en la forma de Representación del Conocimiento que mas se acople a dicha información (sistema experto).
 - b) Crear un banco de preguntas en una base de datos para permitir la autoevaluación teórica de los conocimientos adquiridos, tanto en el aula de clase como en la herramienta.

BIBLIOGRAFÍA

BECERRA, Cesar, Los 600 Principales Métodos del JAVA, Primera Edición, Editorial Kimpres Ltda., Colombia, 1998.

CEBALLOS, Fco Javier; Java 2 Curso de Programación; Editorial Alfaomega S.A. México. 2000.

CORREDOR Montagut, Martha Vitalia. Principios de inteligencia artificial y sistemas expertos. Ediciones UIS. Bucaramanga, 2000.

CORREDOR, Montagut Martha Vitalia, La Educación en Línea: Una Reflexión sobre sus Posibilidades en Educación Superior, Ediciones UIS, Bucaramanga, 2004.

FERNANDEZ Galán, Severino; **GONZALEZ** Boticario, Jesús; **MIRA** Mira, José. Problemas Resueltos de Inteligencia Artificial Aplicada. Búsqueda y Representación. Editorial Addison – Wesley. España, 1998.

FOWLER M. & SCOTT K, UML Gota a Gota, Primera Edición, Editorial Addison Wesley, México, 1999.

GIL Javier, **TEJEDOR** Jorge, **YAGUE** Agustín, **ALONSO** Santiago, **GUTIERREZ** Abraham. Creación de sitios Web con PHP 4. Primera Edición. Editorial McGraw Hill. México 2001.

KOWALSKI, Robert, Lógica, Programación e Inteligencia Artificial. Ediciones Diaz de Santos S.A, Primera Edición, Madrid. 1986.

LARMAN, Craig. UML y Patrones, introducción al análisis y diseño orientado a objetos, primera edición,. Prentice Hall, 1999.

NIL. J Nilsson, Principios de Inteligencia Artificial, Ediciones Diaz de Santos S.A, Madrid.

PRESSMAN, Roger S. Ingeniería del Software - Un enfoque práctico. Cuarta Edición. Editorial McGraw Hill. México. 1998.

RICH Elaine, **KNIGHT** Kevin, Inteligencia Artificial. Segunda Edición. Editorial McGraw Hill. Madrid. 1994.

RUSELL, Stuart. Inteligencia artificial un enfoque moderno. Editorial Prentice Hall Hispanoamericana.S.A. Primera edición. México, 1996.