



**Escuela Superior Politécnica de Chimborazo**

**FACULTAD DE INFORMATICA Y ELECTRONICA**

**PROYECTO**

**FORMACION DE INGENIERIA EN SISTEMAS INFORMATICOS**

**Estudio de las Técnicas de Inteligencia Artificial  
mediante el apoyo de un Software Educativo**

**TESIS DE GRADO**

**Previa a la Obtención del Título de Ingeniero en  
Sistemas Informáticos**

**PRESENTADO POR:**

**José Raúl Salao Bravo**

**Riobamba –Ecuador**

**2009**

## **AGRADECIMIENTOS**

Le hago llegar mis más sinceros agradecimientos a todas aquellas personas que de una forma u otra posibilitaron que este trabajo se hiciera realidad.

- A mis tutores por su confianza, sus grandes ideas, su colaboración y empeño en sacar adelante este proyecto.
- A mis profesores, los cuales desde el inicio de mi carrera me enseñaron nuevos caminos, aportándonos grandes conocimientos y fuerza creadora.
- A mis compañeros por su gran confianza, amistad y ayuda diaria.
- A mis queridos padres y hermanos por educarnos y guiarnos siempre por el camino correcto.

## **DEDICATORIA**

*Dedico este trabajo a todas aquellas personas que hicieron posible de una forma u otra la realización del mismo, en especial a toda mi familia.*

## **DECLARACIÓN DE AUTORIDAD**

“Declaro que soy el único autor de este Proyecto de Diploma y autorizo a la Universidad de Pinar del Río “Hermanos Saíz Montes de Oca” hacer uso del mismo con la finalidad que estime pertinente”.

---

JOSE RAUL SALAO

Egresado Ingeniería en Sistemas ESPOCH

**NOMBRES****FIRMAS****FECHA**

Dr Ms,C. Romeo Rodríguez .....

DECANO DE LA FACULTAD  
INFORMATICA Y ELECTRONICA

Ing Danilo Pastor .....

DIRECTOR DE PROFESIS

Dr José Quintín Cuador Gil .....

DIRECTOR DE TESIS

Ms Caridad Salazar Alea .....

PRESIDENTE TRIBUNAL

Ms. Ricardo Inouye Rodríguez .....

MIEMBRO

Tlgo. Carlos Rodríguez .....

DIRECTOR DEL CENTRO  
DE DOCUMENTACION

## **Resumen.**

El presente trabajo es un software educativo para el estudio de la materia de Inteligencia artificial, fue desarrollado con la finalidad de conocer más a fondo las principales técnicas, conceptos y aplicaciones que servirán para el aprendizaje de estos temas a los estudiantes de Informática de la Escuela Superior Politécnica de Chimborazo.

En los recursos utilizados pudimos contar con tecnología para la Web utilizando el entorno de programación PHP, MySQL, Apache. Para el diseño utilizamos la herramienta Dreamweaver como editor de páginas Web. Utilizamos técnicas pedagógicas pues es un sistema de aprendizaje.

Mediante este Sistema se accede a la información de Inteligencia Artificial y apoyo a los estudiantes a comprender estos temas para aprender, evaluarse y además mediante un análisis de los resultados se evidenció que un 80% se mejora el aprendizaje de las técnicas y conceptos de Inteligencia Artificial. Brinda una interfaz amigable de navegación, la cual tuvo una gran aceptación y fue comprobada su utilidad mediante la prueba de implementación y puesta en marcha en la Red durante 15 días, al cual accedieron 50 usuarios quienes mediante la aplicación de entrevistas y encuestas permitieron conocer la satisfacción del 90% de la labor desarrollada.

Se concluye que este software educativo de Inteligencia Artificial desarrollado puede ser utilizado para el aprendizaje de esta materia.

Recomendamos utilizar este software en el curriculum de la facultad para el desarrollo de la Inteligencia Artificial en la Escuela de Informática.

## **Summary.**

The present work is an educative software for the study of Artificial intelligence subject, it was developed with the purpose of knowing more the main techniques, concepts and applications that will be used for the learning of these subjects for Computer Science students of Polytechnical Superior School of Chimborazo.

In the used resources we could count on technology for the Web using the surroundings of programming PHP, MySQL, Apache. For the design we used the Dreamweaver tool as a publisher of Web pages. We used pedagogical techniques because it is a learning system.

By means of this System it is acceded to the information of Artificial intelligence and support to the students to understand these subjects, to learn to evaluate themselves and in addition by means of an analysis of the results we can demonstrate that a 80% improve the learning of the techniques and concepts of Artificial intelligence. It offers a friendly interface of navigation, which had a great acceptance and it was verified its utility by a test of implementation and by the beginning in the Network during 15 days, which involved 50 users who acceded at the application of interviews and surveys, they allowed to know the satisfaction of a 90% of developed work.

We conclude that this educative software of Artificial intelligence can be used for the learning of this subject

We recommended to use this software in the curriculum of the Faculty for the development of the Artificial intelligence in the School of Computer Science.

# INDICE

INTRODUCCIÓN

PROBLEMÁTICA.

JUSTIFICACIÓN DEL PROYECTO DE TESIS DE GRADO

OBJETIVOS

PLAN GENERAL DE TRABAJO

MATERIALES Y MÉTODOS.

CAPÍTULO I. ESTADO DEL ARTE

<b>1.1 INTRODUCCIÓN.</b> -----	<b>8 -</b>
<b>1.2 UNIVERSIDAD DE PINAR DEL RÍO HERMANOS SAÍZ MONTES DE OCA.</b> -----	<b>9 -</b>
<b>1.3 SISTEMA DE ESTUDIO PLAN D</b> -----	<b>11 -</b>
1.3.1 ASIGNATURA DE INTELIGENCIA ARTIFICIAL. -----	12 -
<b>1.4 FUNDAMENTOS DE INTELIGENCIA ARTIFICIAL</b> -----	<b>14 -</b>
<b>1.5 CONCEPTO DE INTELIGENCIA ARTIFICIAL</b> -----	<b>14 -</b>
<b>1.6 EVOLUCIÓN HISTÓRICA.</b> -----	<b>15 -</b>
<b>1.7 APLICACIONES DE LA IA.</b> -----	<b>18 -</b>
<b>1.8 ÁREAS DE LA INTELIGENCIA ARTIFICIAL (I.A.)</b> -----	<b>19 -</b>
1.8.1 REDES NEURONALES ARTIFICIALES (RNA).-----	19 -
1.8.2 ALGORITMOS GENÉTICOS (AG).-----	20 -
1.8.3 SISTEMAS EXPERTOS (SS.EE.). -----	20 -
<b>1.9 HERRAMIENTAS DE DESARROLLO DEL SOFTWARE TUTORIAL I.A.</b> -----	<b>21 -</b>
1.9.1 PHP. -----	22 -
1.9.2 APACHE.-----	23 -
1.9.3 MYSQL.-----	23 -



<b>1.10 CONCLUSIONES.</b>	<b>25</b>
---------------------------	-----------

## CAPÍTULO II. REPRESENTACIÓN DEL CONOCIMIENTO

<b>2.1 INTRODUCCIÓN.</b>	<b>26</b>
--------------------------	-----------

<b>2.2 CONOCIMIENTO.</b>	<b>26</b>
--------------------------	-----------

<b>2.3 REPRESENTACIÓN DEL CONOCIMIENTO.</b>	<b>28</b>
---	-----------

2.3.1 LÓGICA SIMBÓLICA FORMAL:	29
--------------------------------	----

2.3.2 FORMAS ESTRUCTURADAS:	29
-----------------------------	----

2.3.1.1 LÓGICA PROPOSICIONAL.	29
-------------------------------	----

2.3.1.2 REPRESENTACIÓN MEDIANTE LÓGICA DE PREDICADOS.	31
---	----

2.3.1.3 REGLAS DE PRODUCCIÓN.	34
-------------------------------	----

2.3.2.1 REDES SEMÁNTICAS	35
--------------------------	----

2.3.2.2 ESTRUCTURAS FRAMES (MARCOS).	37
--------------------------------------	----

2.3.2.3 REPRESENTACIÓN ORIENTADA A OBJETOS.	40
---	----

<b>2.4 CONCLUSIONES.</b>	<b>41</b>
--------------------------	-----------

## CAPÍTULO III. TÉCNICAS DE INTELIGENCIA ARTIFICIAL.

<b>3.1. INTRODUCCIÓN</b>	<b>42</b>
--------------------------	-----------

<b>3.2 QUÉ ES UNA TÉCNICA DE INTELIGENCIA ARTIFICIAL (IA)</b>	<b>42</b>
---	-----------

3.2.1 LÓGICA DIFUSA.	43
----------------------	----

3.2.1.1 CARACTERÍSTICAS.	45
--------------------------	----

3.2.1.2 ETAPAS DE LA LÓGICA DIFUSA.	45
-------------------------------------	----

3.2.1.3 APLICACIONES.	46
-----------------------	----

3.2.2 REDES NEURONALES.	47
-------------------------	----

3.2.2.1 ESTRUCTURA RNA.	49
-------------------------	----

3.2.2.2 CARACTERÍSTICAS Y VENTAJAS.	50
-------------------------------------	----

3.2.2.3 MODELOS.	51
------------------	----

3.2.2.4 ARQUITECTURA DE REDES NEURONALES ARTIFICIALES.	52
--	----

3.2.2.5 APLICACIONES.	54
3.2.3 ALGORITMOS GENÉTICOS (AG).	55
3.2.3.1 COMPONENTES DE UN AG.	57
3.2.3.2 FUNCIONAMIENTO DE UN AG.	58
3.2.3.3 APLICACIONES TRADICIONALES DE AG.	60
<b>3.3 BÚSQUEDA EN INTELIGENCIA ARTIFICIAL.</b>	<b>60</b>
3.3.1 DEFINICIÓN FORMAL DE LA SOLUCIÓN DE PROBLEMAS EN LA I.A.	62
3.3.1.1 BÚSQUEDA A CIEGA (BLIND SEARCH) O SIN INFORMACIÓN.	64
3.3.1.2 BÚSQUEDA PRIMERO A LO ANCHO (BREADTH-FIRST)	65
3.3.1.3 BÚSQUEDA PRIMERO EN PROFUNDIDAD. (DEPTH-FIRST).	66
3.3.1.4. BÚSQUEDA HEURÍSTICA O CON INFORMACIÓN.	68
3.3.1.5 GENERACIÓN Y PRUEBA. (GENERATE-AND-TEST)	69
3.3.1.6 HILL CLIMBING O ESCALADOR DE COLINAS.	69
3.3.1.7 ALGORITMO BEST-FIRST (PRIMERO EL MEJOR)	70
3.3.1.8 BÚSQUEDA CON SISTEMAS EVOLUTIVOS.	71
<b>3.4 RESUMEN COMPARATIVO DE LAS TÉCNICAS I.A.</b>	<b>72</b>
<b>CONCLUSIONES.</b>	<b>72</b>
<b><u>CAPITULO 4.</u></b>	
<b>4.1. INTRODUCCIÓN.</b>	<b>73</b>
<b>4.2 FUNCIONAMIENTO DEL SISTEMA</b>	<b>73</b>
<b>4.3 DISEÑO DE LA INTERFAZ-USUARIO.</b>	<b>74</b>
<b>4.4 LENGUAJE DE MODELADO UNIFICADO.</b>	<b>77</b>
<b>4.5. CASOS DE USO.</b>	<b>78</b>
<b>4.6. DISEÑO DE LA BASE DE DATOS DEL SISTEMA.</b>	<b>91</b>
<b>4.7. MAPA DE NAVEGACIÓN</b>	<b>94</b>
<b>4.8 MODELO DE DESPLIEGUE.</b>	<b>94</b>

**4.9 ANÁLISIS DE FACTIBILIDAD----- 96 -**

**4.10 IMPLEMENTACIÓN DEL SOFTWARE ----- 99 -**

**4.11. CONCLUSIONES.----- 100 -**

CONCLUSIONES

RECOMENDACIONES

REFERENCIAS BIBLIOGRÁFICAS

ANEXOS

ANEXOS I.1 ÍNDICE DE TABLAS

ANEXOS I.2 ÍNDICE DE FIGURAS

ANEXOS I.3 PARADIGMAS DE INTELIGENCIA ARTIFICIAL

ANEXOS III. 1 TECNICAS DE INTELIGENCIA ARTIFICIAL

ANEXO IV. 1 INTERFAZ DE USUARIO.

ANEXO IV.2: MODELO LÓGICO DE LOS DATOS DEL TUTORIAL DE I.A.

ANEXO IV.3: MAPA DE NAVEGACION

ANEXO IV.4: PRUEBAS DEL SISTEMA

## INTRODUCCIÓN

En el mundo que vivimos, el desarrollo de herramientas tecnológicas se hace presente cada vez más en nuestras actividades. Las aplicaciones informáticas son una de éstas y juegan un papel muy importante ya que, han llegado a ser tan necesario su uso que nos resulta difícil imaginarnos como sería nuestra vida al prescindir de ellas.

En este proyecto mediante un Tutorial se pretende mostrar las bondades de la Inteligencia Artificial (I.A.) como una herramienta de apoyo en la enseñanza y, a la vez, estimular a la comunidad académica a continuar investigando sobre el desarrollo de esta prometedora área. Se presenta un software para la enseñanza aprendizaje de esta materia y que puede ser utilizado para evaluación a los estudiantes de Informática de la Universidad de Pinar del Río.

Dentro del campo de la Informática esta la Inteligencia Artificial, que ahora es muy utilizada en diferentes campos como la medicina, con la utilización de sistemas expertos, en donde mediante un sistema informático se puede hacer consultas y diagnósticos de determinados síntomas de un paciente, lo que nos permitirá conocer las enfermedades en base al conocimiento y reglas almacenadas. Las grandes empresas la utilizan para la fabricación de sus productos en sistemas de reconocimiento de patrones y sistemas de control con lógica difusa, en la educación para el desarrollo de sistemas de aprendizaje.

Existen varias aplicaciones en nuestra vida cotidiana en donde podemos utilizar las técnicas de IA, para con ello lograr mejorar nuestro trabajo ya sea en control, automatización, juegos, etc.

Empresas como Ecuatoriana de Cerámica ubicada en la ciudad de Riobamba, para la fabricación de los distintos modelos de baldosas que produce, utiliza la técnica de Reconocimiento de Patrones, es decir mediante un software se evalúa si determinado producto cumple con el

tamaño, relieve y la superficie de las normas establecidas, con lo que se logra un mejor control de calidad de sus productos lo cual aumenta el rendimiento y productividad.

En la Empresa Parmalat del Ecuador ubicado en Lasso, Latacunga; para el control de temperatura de pasteurización de leche de larga duración se utiliza un sistema informático, el mismo que mediante técnicas de lógica difusa abre y cierra válvulas neumáticas para controlar el ingreso de vapor de un caldero y así mantener la temperatura de trabajo adecuada.

Lo que esperamos del presente trabajo es a través de un software tener conocimiento claro de los temas de Inteligencia Artificial pues la idea que hoy tenemos de esta materia es demasiado compleja, se pretende cambiar esa visión y darnos cuenta que podemos aplicarlo a nuestro beneficio. Así vamos a utilizar la tecnología para nuestro desarrollo y adelanto tecnológico y contribuir con el progreso del país.

## **PROBLEMÁTICA.**

El avance en que se encuentra la Universidad Pinar del Río para fortalecer el estudio independiente nos obliga a trabajar en la búsqueda de alternativas de auto aprendizaje en la asignatura de Inteligencia Artificial. El Plan D de la Facultad de Informática y Telecomunicaciones así lo señala.

Actualmente en la Facultad de Informática y Telecomunicaciones de la Universidad Pinar del Río no se cuenta con una aplicación educativa dirigida al estudio de la Inteligencia Artificial; debido principalmente a la falta de información y complejidad de este tema. El estudio de la Inteligencia Artificial es teórico y complejo por lo que se podría aprovechar de mejor manera las herramientas tecnológicas existentes actualmente.

Este trabajo está centrado en estudiar más a fondo la inteligencia artificial mediante el apoyo de un software educativo que se aplicará para la enseñanza en la Universidad de Pinar del Río, y a la vez ilustra sobre sus aplicaciones en la vida real mediante la utilización de técnicas de Inteligencia Artificial y ejemplos de aplicación, debido a que el estudiante debe aprender a trabajar y aprovechar el tiempo disponible y desarrollar habilidades básicas en el aprendizaje individual y contribuir a que transiten de forma exitosa por su formación como ingenieros.

El objeto de la investigación es la aplicación educativa sobre las técnicas de inteligencia artificial y el campo de acción es el perfeccionamiento de la enseñanza de las técnicas de Inteligencia Artificial en la carrera de Informática en la Universidad de Pinar del Río.

## **JUSTIFICACIÓN DEL PROYECTO DE TESIS DE GRADO**

La Universidad Pinar del Río, de forma general, transita por un proceso de cambio y desarrollo, lo que hace necesario la construcción de un software educativo y la aplicación de tecnología educativa en forma general para el perfeccionamiento de los planes de estudio en caso particular la Inteligencia Artificial y así, aprender más a fondo esta materia con sus diferentes paradigmas, entre los más relevantes las redes neuronales, algoritmos genéticos, sistemas de lógica difusa, con sus diferentes aplicaciones en la vida cotidiana.

Demostrar que mediante el apoyo de un software se puede aprender las Técnicas de Inteligencia Artificial conocer su funcionamiento, aplicaciones y realizar un estudio comparativo entre la utilización de reglas de producción, redes semánticas y lógica difusa.

Actualmente no existe un Sistema de aprendizaje en esta materia, por lo que el uso de un tutorial de Inteligencia Artificial supone varias ventajas. Los alumnos pueden acceder libremente a los equipos en su laboratorio, pudiendo estudiar en un horario flexible, y sin requerir la presencia continua de un profesor. Cada estudiante podrá dedicar el tiempo que estime oportuno para estudiar e incluso podrá realizarla las veces que quiera, posibilitando una mejor asimilación de los conceptos, por otro lado, permite que el alumno pueda conocer en qué medida los conceptos que ha ido adquiriendo se han afianzado.

# **OBJETIVOS**

## **OBJETIVO GENERAL.**

Potenciar el estudio de las técnicas, funcionamiento y aplicaciones de la Inteligencia Artificial mediante el apoyo de una aplicación educativa.

## **OBJETIVOS ESPECÍFICOS:**

- Implementar un tutorial de aprendizaje de Inteligencia Artificial que permita apoyar la solución de la problemática existente.
- Establecer parámetros de comparación para evaluar las diferentes técnicas de Inteligencia Artificial y seleccionar la mejor alternativa para determinada aplicación.
- Implementar el tutorial en la Red de la Facultad para su explotación por parte de profesores y estudiantes.



## **PLAN GENERAL DE TRABAJO**

Para el desarrollo del proyecto de tesis se ha determinado el siguiente plan de trabajo, el cual se ha dividido en las siguientes fases:

En el Capítulo 1 se tratará el estado del arte del tema, aquí se estudia la problemática a resolver, los materiales y herramientas (hardware y software) que utilizamos en este proyecto, se analizará la Inteligencia Artificial empezando con una breve introducción donde se hará un poco de historia y estudio de los conceptos básicos.

En el Capítulo 2 debido a que el desarrollo de sistemas inteligentes requieren una adecuada representación del conocimiento, en este capítulo vamos a analizar el conocimiento, sus formas de representación, características, ejemplos y clasificación: Redes Semánticas, Lógica de Predicados, Marcos, etc.

En el Capítulo 3 después de una breve introducción se explican las diversas técnicas de Inteligencia Artificial, los métodos de búsqueda utilizados para la resolución de problemas, las áreas de aplicación y se terminará con las conclusiones de lo que se aprendió después de una profunda investigación en estos temas.

El Capítulo 4 contiene las fases de desarrollo de software y se escribirá sobre el funcionamiento del Sistema mediante Casos de Uso, Interfaz de Usuario, Diagrama de diseño de la Base de Datos y se cerrará el Capítulo con las conclusiones acerca de la aplicación que se diseñó.

## **MATERIALES Y MÉTODOS.**

Para la realización del presente trabajo se tomó en cuenta la opinión de personal que trabaja dentro de la automatización mediante conversaciones, en foros y seminarios, así como el Análisis Documental de la literatura científica, artículos y sitios Web dedicados al tema de Inteligencia Artificial. Las fuentes de información a consultar que se utilizaron fueron: libros, monografías, sitios especializados y oficiales en la temática, principalmente en Internet.

La implementación de este Sistema, se realizó a través del entorno de programación PHP, APACHE, MySQL. También utilizamos herramientas como DREAMWEAVER, PHOTOSHOP, PROJECT, OFFICE, RATIONAL ROSE.

Los métodos utilizados para cumplir con los objetivos fueron: el inductivo y deductivo, entre las técnicas: la observación y entrevistas.

# **CAPÍTULO I. Estado del arte**

## **1.1 Introducción.**

El estudio de la asignatura de Inteligencia Artificial dentro de la carrera de Ingeniería en Informática es muy importante debido a las aplicaciones que existen para este campo. Actualmente, en varias áreas como la medicina, ingeniería, producción, educación, está presente la utilización de la Inteligencia Artificial a través de sus Técnicas como son: Lógica Difusa, Sistemas Expertos, Redes Neuronales, etc. Así para el estudio de esta materia y otras, la Universidad de Pinar del Río Hermanos Saíz Montes de Oca ha implementado modelos de enseñanza que permitan cumplir los objetivos propuestos y se adapten a nuestra realidad, teniendo siempre como objetivo mejorar el sistema de aprendizaje de los estudiantes.

En la investigación realizada, tanto en esta Universidad como en otras, así como en la red de redes, hemos encontrado múltiples bibliografías, algunos trabajos en páginas Web así como textos en copia dura que abordan de una forma u otra el tema a tratar. En ningún caso hemos encontrado una aplicación que contribuya, en la forma que nos planteamos, a la solución de la problemática en cuestión.

Todo ello ratifica la necesidad y utilidad de la creación de nuestra aplicación.

El inicio de este capítulo trata sobre la problemática a resolver con este trabajo, vamos a investigar la Misión y Visión de la Facultad de Informática y Telecomunicaciones, pues esta tesis está dedicada a contribuir con mejorar el aprendizaje de los estudiantes de la materia de Inteligencia Artificial mediante un Tutorial. Luego vamos a analizar el funcionamiento de este software como herramientas de programación, entorno de desarrollo y la introducción a los conceptos básicos de la Inteligencia Artificial.

## **1.2 Universidad de Pinar del Río Hermanos Saíz Montes de Oca.**

La Universidad de Pinar del Río Hermanos Saíz Montes de Oca ha seguido los perfeccionamientos de los planes y programas de estudio dentro de la Educación Superior, para formar profesionales con una sólida preparación y con conocimientos esenciales de su profesión y habilidades profesionales de mayor generalidad que los dotasen con la capacidad necesaria para responder al gran reto que el país tenía por delante, en el curso 1990-1991 se hace vigente la aplicación de los Planes de Estudio "C".

Estos programas concibieron lograr una formación de perfil amplio en el pregrado, mediante una mayor integración de los componentes académico, investigativo y laboral, egresando en mejores condiciones para adquirir y actualizar conocimientos en el postgrado y para desarrollar las nuevas habilidades que requieren las diversas y crecientes demandas de la dinámica del mundo del trabajo y el avance de la sociedad.

El actual Departamento de Informática, anteriormente de Computación, surgió en el año 1987, asociado al Departamento de Matemática de la entonces Facultad de Economía, no es hasta el 1990 que se independiza como departamento básico pasando a la Vicerrectoría de Investigaciones y Postgrado hasta el año 1998, pasando de nuevo a la Facultad de Economía y, en el 2000, pasa a la Facultad de Ciencias Técnicas donde se comienza a preparar el expediente para solicitar al Ministerio de Educación Superior la apertura de la Carrera de Informática,

aprobándose la misma para su inicio en el curso 2001-2002 y por tal motivo se crea en este curso la Facultad de Informática y Telecomunicaciones, cambiándose el nombre al Departamento por Departamento de Informática.

Desde su creación, en 1987 ha impartido todas las asignaturas de Computación de las Carreras de la Universidad Pinar del Río (UPR), la responsabilidad de la superación de postgrado de la propia Universidad y del territorio, las asignaturas de Informática al crearse la Universalización y la Informatización de la propia Universidad y de la Sociedad Pinareña.

Actualmente el reto es el PLAN DE ESTUDIOS "D", y dentro de sus principales objetivos están:

- Desarrollar las capacidades para el aprendizaje autónomo y colaborativo a través del modo de asimilación de los contenidos, donde predomine un enfoque sistémico con tendencia hacia niveles de asimilación productivos.
- Desarrollar y consolidar a través de estrategias de estudio independiente y de trabajo grupal valores éticos y morales.

Este trabajo pretende contribuir con la Misión y Visión de la Facultad de Informática y Telecomunicaciones de la Universidad de Pinar del Río [Fitweb, 2008].

### **Misión**

Contribuir al desarrollo y aplicación de las tecnologías de la información y las comunicaciones en el territorio mediante la formación integral de profesionales de la Informática, las Telecomunicaciones y la Electrónica, así como su superación posgraduada y al avance de la ciencia y la técnica en correspondencia con las necesidades y expectativas de la Sociedad Pinareña y nuestro Estado Revolucionario.

## **Visión**

Se consolida el trabajo por lograr altos niveles en formación integral de los estudiantes de los diferentes tipos de cursos evidenciados a través de la existencia de cultura universitaria en ambas carreras en lo referente al movimiento político por elevar la calidad del profesional revolucionario.

La Facultad contribuye con protagonismo a la gestión en la Universidad Pinar del Río (UPR) de la informatización, además se cuenta con el equipamiento indispensable para la formación de un profesional competitivo, se dispone de una importante fuente de información en una Intranet fiable, con las informaciones más necesarias y actualizadas en sitios FTP y servidores.

### **1.3 Sistema de Estudio Plan D**

El Plan de Estudios de la carrera de Ingeniería Informática desde su último diseño en la variante de Planes C, se ha caracterizado por un conjunto apreciable de modificaciones causadas en primera instancia por el enriquecimiento y la variación del objeto de estudio en sí, que ha motivado ajustes relacionados con el entorno universitario local y las relaciones con las entidades productivas y de servicio. Así, el diseño del Plan de Estudios D se fundamenta en las transformaciones que han acontecido en el país y que impusieron el perfeccionamiento de los planes de estudio, con un énfasis marcado en los siguientes aspectos:

La universalización de la educación superior y los Programas de la Revolución, relacionado con la creación del Programa de formación emergente de maestros primarios.

- Fortalecimiento de los Institutos Politécnicos de Informática.
- Las tendencias internacionales en el desarrollo informático y su relación con otras tecnologías emergentes y en la enseñanza universitaria cubana [plan D, 2008].

### **1.3.1 Asignatura de Inteligencia Artificial.**

#### **Objetivos Educativos**

- Ampliar la capacidad de razonamiento, el pensamiento lógico y el nivel de abstracción necesarios en la asignatura.
- Aplicar capacidades cognoscitivas, de razonamiento y de las formas del pensamiento lógico, hábitos de lectura y trabajo independiente.
- Desarrollar el espíritu de auto superación que permita mantenerse actualizado en los avances de los temas de la asignatura.

#### **Objetivos Instructivos**

- Determinar la información y el conocimiento a procesar en organizaciones de base productivas y de servicio considerando los formalismos de representación de conocimientos y las técnicas para su adquisición o generación.
- Diseñar e implementar sistemas que utilicen técnicas de IA para el control del funcionamiento y la toma de decisiones en organizaciones de base.
- Ejercer un liderazgo eficaz frente a diferentes grupos durante el desarrollo y utilización de sistemas informáticos inteligentes.
- Manejar eficazmente diferentes estrategias de comunicación que viabilicen el trabajo en grupos diferentes y con personas y profesionales de diferentes carreras al desarrollar y aplicar sistemas inteligentes.

#### **Conocimientos básicos a adquirir.**

Conceptos básicos sobre utilización de árboles para la representación de estados en el proceso de búsqueda. Búsqueda a ciegas y heurísticas. Búsquedas con adversarios. Conceptos básicos sobre reglas de producción. Sistemas Expertos y Sistemas de Reglas de Producción. Tratamiento de la incertidumbre. Sistemas difusos. Redes Semánticas, Marcos ("Frames") y Ontologías para la representación del conocimiento.

Introducción a la Minería de Datos, Metodologías. Redes Neuronales. Razonamiento Basado en Casos.

En el presente plan D se determinan los conocimientos esenciales de la disciplina que todo egresado de esta especialidad debe conocer y las habilidades asociadas que son requeridas, creándose a partir de ello las asignaturas obligatorias que se integran en el currículo base de la carrera.

Aquellos conocimientos que hasta cierto punto constituyen especializaciones en el área de la Inteligencia Artificial, se consideraron apropiados para ser desarrollados en temas optativos y se proponen algunos de ellos, aunque no se agotan todas las posibilidades que podrían ser consideradas.

### **Trabajo independiente del estudiante**

Las habilidades para el trabajo independiente deberán comenzar a desarrollarse en los estudiantes desde su ingreso a la Educación Superior, por lo que el curso introductorio para los estudios de Ingeniería, de al menos seis semanas de duración, que se llevará a cabo al inicio de la carrera tiene entre sus contenidos esenciales la asignatura Aprender a Aprender. Esta tiene el propósito de desarrollar habilidades básicas en el aprendizaje individual y contribuir a que transiten de forma exitosa por su formación como ingenieros.

El sistema de enseñanza se concibe para que el estudiante pueda apropiarse de los conocimientos incluidos en el programa de cada asignatura mediante el análisis y estudio de casos prácticos, seminarios, laboratorios y auto aprendizaje y con la realización de proyectos de curso y trabajos en las Prácticas Profesionales. Se utiliza como principio de formación el aprender haciendo, aplicando las mejores prácticas para desarrollar y gestionar software.

Ambas disciplinas integradoras, coordinan su labor en las dimensiones curricular y profesional-investigativa, para que de su sinergia lograr la



integración y fortalecimiento de las principales habilidades, conocimientos y valores a formar en el estudiante.

#### **1.4 Fundamentos de Inteligencia Artificial**

La I.A. (Inteligencia Artificial) es una de las disciplinas más nuevas. Formalmente se inicia en 1956 cuando se acuñó este término, sin embargo el estudio de la inteligencia contemplada como el razonamiento humano viene siendo estudiado por los filósofos hace más de 2 milenios.

##### **Consideraciones Generales:**

En primer lugar, revisemos algunas definiciones generales de inteligencia, antes de intentar definir inteligencia artificial.

Inteligencia es la aptitud de crear relaciones. Esta creación puede darse de manera puramente sensorial, como en la inteligencia animal; también puede darse de manera intelectual, como en el ser humano, que pone en juego el lenguaje y los conceptos. También se la puede conceptualizar como la habilidad para adquirir, comprender y aplicar conocimiento; o la aptitud para recordar, pensar y razonar [Choque, 2002].

#### **1.5 Concepto de Inteligencia Artificial**

El concepto de Inteligencia Artificial (I.A.) puede tener diferentes connotaciones, dependiendo de la fuente que se tome como referencia y del punto de vista de esta. Lo que sí es claro es que la I.A. tiene como uno de sus objetivos el estudio del comportamiento inteligente de las máquinas [Acedo, 2005].

La Inteligencia Artificial es el estudio de cómo hacer que los ordenadores hagan cosas que por el momento son realizadas mejor por los seres humanos [Choque, 2002].

**Presenta notables diferencias con la Informática Clásica.**

	<b>INFORMÁT. "CLÁSICA"</b>	<b>I.A.</b>
<b>DATOS</b>	Numéricos o alfanuméricos	Simbólicos. Ideas y conocimiento.
<b>PROCESO</b>	Algoritmos deterministas	Tipo lógico-abierto. Búsqueda heurística
<b>RESULTADO DEL PROCESO</b>	Exacto	Satisfactorio
<b>REPRESENT. DEL CONOCIMIENTO</b>	Implícita al programa	Explícita. Separación entre base de conocimientos y mecanismos de control
<b>MÉTODO DE PROGRAMAC.</b>	"CÓMO" hacer	"QUÉ" HACER
<b>TÉCNICAS</b>	Algoritmos. Tipos Abstractos de datos.	Búsqueda inteligente. Representación. del conocimiento.

**Tabla I.1 Informática clásica vs Inteligencia Artificial**

**1.6 Evolución Histórica.**

La Inteligencia Artificial "nació" en 1943 cuando Warren McCulloch y Walter Pitts propusieron un modelo de neurona del cerebro humano y animal. Estas neuronas nerviosas abstractas proporcionaron una representación simbólica de la actividad cerebral [Wikimedia, 2007].

En 1950, Turing publicó en la revista Mind el artículo Computing Machinery and Intelligence en el que introducía el concepto de Test de Turing. Este artículo puede considerarse el precursor de muchos de los desarrollos actuales en el campo de la Inteligencia Artificial. El test consistía en juzgar el nivel de inteligencia de una máquina. Se supone un juez situado en una habitación, y una máquina y un ser humano en otras. El juez debe descubrir cuál es el ser humano y cuál es la máquina, estándoles a los dos permitidos mentir al contestar por escrito las preguntas que el juez les hiciera. La tesis de Turing es que si ambos jugadores eran suficientemente hábiles, el juez no podría distinguir quién era el ser humano y quién la máquina [ABCdatos, 2008].

En el año 1955 Herbert Simón, el físico Allen Newell y J.C. Shaw, programador de la RAND Corp. y compañero de Newell, desarrolla el primer lenguaje de programación orientado a la resolución de problemas de la Inteligencia Artificial, el IPL-11. Un año más tarde estos tres científicos desarrollan el primer programa de Inteligencia Artificial al que llamaron Logic Theorist, el cual era capaz de demostrar teoremas matemáticos, representando cada problema como un modelo de árbol, en el que se seguían ramas en busca de la solución correcta, que resultó crucial. Este programa demostró 38 de los 52 teoremas del segundo capítulo de Principia Mathematica de Russel y Whitehead.

En 1956, con la ahora famosa conferencia de Dartmouth, organizada por John McCarthy y en la cual se utilizó el nombre de inteligencia artificial para este nuevo campo, se separó la Inteligencia Artificial de la ciencia del computador, como tal. Se estableció como conclusión fundamental la posibilidad de simular inteligencia humana en una máquina.

En 1957 Newell y Simon continúan su trabajo con el desarrollo del General Problems Solver (GPS). GPS era un sistema orientado a la resolución de problemas; a diferencia del Logic Theorist, el cual se orientó a la demostración de teoremas matemáticos, GPS no estaba programado para resolver problemas de un determinado tipo, razón a la cual debe su nombre. Este sistema a pesar de que pudo resolver determinados problemas no tuvo el éxito esperado.

En 1957 McCarthy desarrolló el lenguaje LISP. La IBM contrató un equipo para la investigación en esa área y el gobierno de USA aportó dinero al MIT también para investigación en 1963.

Entre los últimos años de la década del 50 y principios de los años 60 Robert K. Lindsay, desarrolla un programa orientado a la lectura de oraciones en inglés y la obtención de conclusiones a partir de su interpretación, este programa que se denominó Sad Sam, marco un hito en una de las ramas de la Inteligencia Artificial, el Procesamiento del Lenguaje Natural. Este sistema podía leer oraciones del tipo "Jim es

hermano de John" y "La madre de Jim es Mary", a partir de ella el sistema concluía que Mary debía ser también la madre de John.

En el mismo período de tiempo hay trabajos importantes de Herbert Gelernter, de IBM, quien desarrolla un "Demostrador Automático de Teoremas de la Geometría", Alex Bernstein desarrolla un programa para el juego de ajedrez que se considera el antecedente para "Deep Blue".

En 1961 se desarrolla SAINT (Symbolic Automatic Integrator) por James Slagle el cual se orienta a la demostración simbólica en el área del álgebra. En 1964 Bertrand Raphael construye el sistema SIR (Semantic Information Retrieval) el cual era capaz de comprender oraciones en inglés.

En los primeros años de la década del 60 Frank Roseblatt desarrolla, en la Universidad de Cornell, un modelo de la mente humana a través de una red neuronal y produce un primer resultado al cual llama perceptrón. Este sistema era una extensión del modelo matemático concebido por McCulloch y Pitts para las neuronas, y funcionaba basándose en el principio de "disparar" o activar neuronas a partir de un valor de entrada el cual modifica un peso asociado a la neurona, si el peso resultante sobrepasa un cierto umbral la neurona se dispara y pasa la señal a aquellas con las que está conectada. Al final, en la última capa de neuronas, aquellas que se activen definirán un patrón el cual sirve para clasificar la entrada inicial.

Este trabajo constituye la base de las redes neuronales de hoy en día, sin embargo a raíz de su desarrollo sufrió fuertes críticas por parte de Marvin Minsky y Seymour Papert lo cual provocó que la mayoría de los investigadores interesados en el tema lo abandonaran, y este no se retomara hasta los años 80.

En 1965-70, comenzaron a aparecer los programas expertos, que predicen la probabilidad de una solución bajo un set de condiciones, entre esos proyectos estuvo: DENDRAL, que asistía a químicos en estructuras químicas complejas euclidianas; MACSYMA, producto que asistía a

ingenieros y científicos en la solución de ecuaciones matemáticas complejas, etc.

En la década 1970-80, creció el uso de sistemas expertos, muchas veces diseñados para aplicaciones médicas y para problemas realmente muy complejos como MYCIN, que asistió a médicos en el diagnóstico y tratamiento de infecciones en la sangre. Otros son: R1/XCON, PIP, ABEL, CASNET, PUFF, INTERNIST/CADUCEUS, etc. Algunos permanecen hasta hoy.

De 1975 en adelante, comienza la era de los lenguajes expertos (shells) como EMYCIN, EXPERT, OPSS, etc. para luego tratar de que éstos sean más amigables y funcionales.

En 1980 la historia se repitió con el desafío japonés de la quinta generación de computadoras, que dio lugar al auge de los sistemas expertos pero que no alcanzó muchos de sus objetivos, por lo que este campo sufrió una nueva interrupción en los años noventa.

En la actualidad se está tan lejos de cumplir la prueba de Turing como cuando se formuló: Existirá Inteligencia Artificial cuando no seamos capaces de distinguir entre un ser humano y un programa de computadora en una conversación a ciegas.

### **1.7 Aplicaciones de la IA.**

- Tratamiento de Lenguajes Naturales: Capacidad de Traducción, Órdenes a un Sistema Operativo, Conversación Hombre-Máquina, etc.
- Sistemas Expertos: Sistemas que se les implementa experiencia para conseguir deducciones cercanas a la realidad.
- Problemas de Percepción: Visión y Habla, reconocimiento de voz, obtención de fallos por medio de la visión, diagnósticos médicos, etc.
- Aprendizaje: Modelización de conductas para su implante en computadoras.
- Juegos: ajedrez, damas, del tipo estratégicos.

- Matemáticas: cálculo simbólico, demostración.
- Ingeniería: diseño, detección de fallos, planificación.

### **1.8 Áreas de la Inteligencia Artificial (I.A.)**

Estas áreas o campos hacen referencia a las subtareas que se engloban dentro de la I.A. y dependiendo del paradigma utilizado para el desarrollo del modelo se clasifican de diferente manera:

En el Paradigma Conexionista o Subsimbólico los más reconocidos son las Redes Neuronales Artificiales y los Algoritmos Genéticos.

Por otro lado, en el Paradigma Simbólico se encuentran los Sistemas Expertos, los programas de resolución de problemas matemáticos o los programas de juegos. Sin embargo existen muchas otras técnicas utilizadas en la I.A. pudiendo caer en discusión su clasificación. A continuación se describen brevemente las técnicas más significativas y de mayor aplicación en la actualidad.

#### **1.8.1 Redes Neuronales Artificiales (RNA).**

Al inicio de la IA, el conocimiento se representaba usando reglas, redes semánticas, modelos probabilísticos, etc. La metodología para la resolución de problemas era el uso del proceso humano de razonamiento lógico relacionando las causas del problema con los conocimientos que se poseían y luego se procesaba la información secuencialmente.

Al avanzar la IA, aparecieron problemas complejos en los que no era posible representar el conocimiento de manera explícita ni procesarlo utilizando el razonamiento lógico. La posible solución a este inconveniente era utilizar estructuras computacionales paralelas basadas en redes neuronales biológicas; así surgen las redes neuronales artificiales.

Las RNA se estructuran como un grupo numeroso de procesadores conectados, entre sí y trabajando en paralelo [TREC, 2001].

### **1.8.2 Algoritmos Genéticos (AG).**

Los AG se han desarrollado con la intención de optimizar poblaciones o conjuntos de soluciones posibles de un determinado problema mediante la utilización de la genética biológica. La genética es la ciencia biológica que estudia la variabilidad y la herencia de los seres vivos analizando el material genético a partir del cual nace todo ser vivo; este material genético se compone del ADN (ácido desoxirribonucleico) que se auto duplica en cada división celular. Esta ciencia es la base para los AG ya que estos intentan, de alguna manera, representar un dato (que puede ser una posible solución) mediante una cadena de bits la cual, en analogía con la genética, sería un cromosoma.

Una vez representados todos los datos (posibles soluciones) se cruzan, mutan o recombinan entre sí, con lo cual se obtienen nuevos datos o cromosomas que pueden o no ser mejores que los anteriores. De ser mejores, este paso se repite hasta la obtención de nuevos cromosomas cuya mejora con respecto a los anteriores sea insignificante por lo que el proceso de evolución concluye con un nuevo grupo o población de datos (en la mayoría de los casos se intenta llegar a una sola solución, dato o cromosoma) que son interpretados como las mejores soluciones posibles del problema que se ha intentado solucionar. [Galeón, 1994].

### **1.8.3 Sistemas Expertos (SS.EE.).**

Se puede decir que los SS.EE. son sistemas compuestos por dos partes componentes, principalmente; una llamado Motor de Inferencias y la otra Base de Conocimientos. En la primera se representa el procedimiento lógico que sigue un experto en determinada área del conocimiento y en la segunda se representa el propio conocimiento del experto, pudiendo ser uno sólo o un grupo de expertos dentro de la misma área, inclusive pueden ser libros especializados en el tema de interés [Warianos, 2008].

Este tipo de sistemas se desarrolla con los objetivos de imitar la inteligencia humana, en este caso de un experto, y de tener asistencia de un experto en cualquier lugar en poco tiempo con respecto al tiempo que

tarda en formarse un profesional humano incluyendo el tiempo de experiencia.

Uno de los requisitos para realizar un SS.EE. es poseer el conocimiento de uno o varios expertos con amplia experiencia en el campo del conocimiento que les compete, ya que es la experiencia quien agrega valor al conocimiento. Un SS.EE. que se basa en los conocimientos de un novato poco puede aportar a un profesional que, si bien, no es un experimentado tampoco es un novato, pero que necesita la asistencia o los conocimientos de un experto que lo guíe en su búsqueda de una solución a un problema complejo o nuevo para el profesional. Existen diferentes tipos de SS.EE. clasificados según el modelo que utilizan, siendo algunos de éstos los siguientes:

- SS.EE. basados en Reglas.
- SS.EE. basados en Probabilidad.
- SS.EE. basados en Redes Neuronales Artificiales.
- SS.EE. basados en Lógica Difusa.

### **1.9 Herramientas de desarrollo del software Tutorial I.A.**

En este punto se presentan las características del software a utilizar para el desarrollo del sistema, que por su facilidad, flexibilidad y mejor manejo se ha elegido:

- Como servidor Web: Apache 2.2.4
- Lenguaje para servidor Web: PHP versión 5.2.3
- Manejador de base de datos: MySQL 5.0.45

Se instalará Apache como servidor Web, lo cual resulta favorable con el uso del lenguaje PHP, pues está principalmente diseñado para correr en este servidor. A su vez, PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y por ODBC a todas las bases de datos posibles en sistemas Microsoft, a partir de las cuales se podrá editar el contenido de la bases de datos del sistema con relativa sencillez.



Esta interacción se realiza, por un lado, a partir de las funciones que PHP propone para cada tipo de base de datos y, por otro, estableciendo un diálogo a partir de un idioma universal: SQL (Structured Query Language), el cual es común a todas las bases de datos, razón por lo cual se utilizará MySQL para dicha interacción.

### **1.9.1 PHP.**

Es un acrónimo recursivo que significa "PHP Hypertext Pre-processor". Publicado bajo la PHP License, está considerado como un software libre. PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo Web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor Web, tomando el código en PHP como su entrada y creando páginas Web como salida. Puede ser desplegado en la mayoría de los servidores Web y en casi todos los sistemas operativos y plataformas sin costo alguno.

#### **Ventajas.**

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos. También PHP permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

Los principales usos del PHP son los siguientes: Programación de páginas Web dinámicas, habitualmente en combinación con el motor de base de datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión [García J,2004].

### 1.9.2 APACHE.

Como servidor Web utilizaremos el modulo Apache, el cual es el más popular y es un servidor de red para el protocolo HTTP (Hyper Text Tranfer Protocol), elegido para poder funcionar como un proceso independiente, sin que solicite el apoyo de otras aplicaciones. [Apache, 2008].

Las Principales razones que justifican la gran aceptación de Apache son:

**Multiplataforma:** Corre en una multitud de sistemas Operativos, lo que lo hace prácticamente universal.

**Libre:** Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor lo podemos saber, sin ningún secreto sin ninguna puerta trasera.

**Configurable:** Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache.

**Multilinguaje:** Apache soporta gran cantidad de lenguajes PERL, PHP, Java. Teniendo todo el soporte que se necesita para tener páginas dinámicas.

Estas y otras características han convertido al servidor Web Apache en el más utilizado actualmente, avalado por compañías élite como Google y superando ampliamente a sus similares en lo que a seguridad y agilidad se refiere.

### 1.9.3 MySQL.

Es un sistema de gestión (o manejador) de bases de datos. Una Base de Datos es una colección estructurada de datos. Puede ser, desde una

simple lista de artículos a las inmensas cantidades de información en una red corporativa.

MySQL es un software de código abierto. Esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. Se puede descargar MySQL desde Internet y usarlo sin pagar nada, de esta manera cualquiera puede inclinarse a estudiar el código fuente y cambiarlo para adecuarlo a sus necesidades.

MySQL usa el GPL (GNU General Public License) para definir que se puede o no hacer con el software en diferentes situaciones.

MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños, aunque esta rapidez es a costa de no implementar ciertos aspectos del SQL.

#### **Principales características:**

- El principal objetivo de MySQL es velocidad y robustez.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y sistemas operativos distintos.
- Sistema de contraseñas y privilegios muy flexible y seguro.
- Todas las palabras viajan encriptadas en la red.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintos idiomas.
- Todos los comandos tienen -help o -? para las ayudas.

### **1.10 Conclusiones.**

- Como conclusiones de este capítulo aprendimos la importancia que tiene el desarrollo de este trabajo para la enseñanza de la materia de Inteligencia Artificial en la Universidad de Pinar del Río, así como todos los detalles para el desarrollo del software que se utilizará para construir esta aplicación.
- En este capítulo se profundizó en el conocimiento de enseñanza del plan D de la Universidad de Pinar del Río. Se abordaron las propuestas de desarrollo de estos temas. Se dedicó a estudiar los conceptos básicos de Inteligencia Artificial, su evolución histórica desde sus inicios hasta nuestros días y sus principales aplicaciones.
- Se analizaron los conceptos básicos de los paradigmas I.A. como redes neuronales, algoritmos genéticos y sistemas expertos, además se muestra la necesidad y las posibilidades para desarrollar un software de enseñanza que permita la difusión de esta materia.

## **CAPÍTULO II. Representación del Conocimiento**

### **2.1 Introducción.**

La Representación del Conocimiento constituye el núcleo de la ciencia de las computadoras. Cada pieza de código escrita es significativa para la representación de algún objeto. En la actualidad la representación del conocimiento es investigada de tal forma que cualquier libro que trate sobre Inteligencia Artificial la considerará un resultado central.

En general una representación debe tener dos capacidades: por un lado una expresividad adecuada y por otro una eficiencia de razonamiento. La expresividad y el razonamiento le confieren al formalismo la capacidad adecuada para ser considerado como una alternativa útil para la representación [Morales, 1999].

El objetivo de este capítulo, es brindar los conceptos básicos de la forma de representación del conocimiento, su clasificación, utilización y distintas características para aprender a utilizarlo en alguna aplicación práctica en donde utilicemos las Técnicas de Inteligencia Artificial.

### **2.2 Conocimiento.**

Puede ser definido como el conjunto de hechos y principios acumulados por la humanidad, o el acto, hecho o estado de conocer. Es la familiaridad con el lenguaje, conceptos, procedimientos, reglas, ideas, abstracciones, lugares, costumbres y asociaciones.

Los conceptos de conocimiento e inteligencia están íntimamente ligados. La inteligencia requiere de la posesión y acceso al conocimiento. Conocimiento no debe ser confundido con datos o información. El conocimiento incluye y requiere del uso de datos e información. Además, combina relaciones, dependencias, y la noción del saber con datos e información.

El conocimiento puede ser de tipo procedimental, declarativo o heurístico.

**Conocimiento procedimental** es aquel conocimiento compilado que se refiere a la forma de realizar una cierta tarea (el saber cómo hacerlo). Por ejemplo, los pasos necesarios para resolver una ecuación algebraica son expresados como conocimiento procedimental [Choque, 2002].

Los modelos procedimentales y sus esquemas de representación almacenan conocimiento en la forma de cómo hacer las cosas. Pueden estar caracterizados por gramáticas formales, usualmente implantadas por sistemas o lenguajes procedimentales y sistemas basados en reglas (sistemas de producción).

Por otro lado, el **conocimiento declarativo** es conocimiento pasivo, expresado como sentencias acerca de los hechos del mundo que nos rodea (el saber que hacer). La información personal en una base de datos es un típico ejemplo de conocimiento declarativo. Tales tipos de datos son piezas explícitas de conocimiento independiente.

El conocimiento declarativo puede ser representado con modelos relacionales y esquemas basados en lógica. Los modelos relacionales pueden representar el conocimiento en forma de árboles, grafos o redes semánticas. Los esquemas de representación lógica incluyen el uso de lógica proposicional y lógica de predicados.

El **conocimiento heurístico** es un tipo especial de conocimiento usado por los humanos para resolver problemas complejos. El adjetivo heurístico significa medio para descubrir. Está relacionado con la palabra griega heuriskein que significa descubrir, encontrar. Se entiende por heurístico a

un criterio, estrategia, método o truco utilizado para simplificar la solución de problemas. El conocimiento heurístico usualmente se lo adquiere a través de mucha experiencia.

### **2.3 Representación del Conocimiento.**

Dado que el conocimiento es importante y primordial para el comportamiento inteligente, su representación constituye una de las máximas prioridades de la investigación en IA [FortuneCity, 2008].

El conocimiento puede ser representado como imágenes mentales en nuestros pensamientos, como palabras habladas o escritas en algún lenguaje, en forma gráfica o en imágenes, como cadenas de caracteres o colecciones de señales eléctricas o magnéticas dentro de un computador.

La forma de representación que se escoja dependerá del tipo de problema a ser resuelto y de los métodos de inferencia disponibles.

Una representación del conocimiento puede ser un esquema o dispositivo utilizado para capturar los elementos esenciales del dominio de un problema. Una representación manipulable es aquella que facilita la computación. En representaciones manipulables, la información es accesible a otras entidades que usan la representación como parte de una computación.

Debido a la variedad de formas que el conocimiento puede asumir, los problemas involucrados en el desarrollo de una representación del conocimiento son complejos, interrelacionados y dependientes del objetivo. En términos generales, se debe tratar que el conocimiento esté representado de tal forma que:

- Capture generalizaciones.
- Pueda ser comprendido por todas las personas que vayan a proporcionarlo y procesarlo.
- Pueda ser fácilmente modificado.

- Pueda ser utilizado en diversas situaciones aún cuando no sea totalmente exacto o completo.
- Pueda ser utilizado para reducir el rango de posibilidades que usualmente debería considerarse para buscar soluciones.

Las representaciones declarativas son usualmente más expansivas y costosas, en el sentido que la enumeración puede ser redundante e ineficiente. Sin embargo, la modificación de las representaciones declarativas es usualmente muy fácil; simplemente se agrega o se elimina conocimiento. Las representaciones procedimentales, en cambio, pueden ser más compactas, sacrificando flexibilidad. Representaciones prácticas pueden incluir elementos tanto declarativos (listado de hechos conocidos), como procedimentales (un conjunto de reglas para manipular los hechos).

La ingeniería cognoscitiva ha adaptado diversos sistemas de representación del conocimiento que, implantados en un computador, se aproximan mucho a los modelos elaborados por la psicología cognoscitiva para el cerebro humano. Entre los principales se tienen:

### **2.3.1 Lógica Simbólica Formal:**

- Lógica proposicional
- Lógica de predicados.
- Reglas de producción.

### **2.3.2 Formas Estructuradas:**

- Redes Semánticas.
- Estructuras marco.
- Representación orientada a objetos.

#### **2.3.1.1 Lógica Proposicional.**

La lógica proposicional es la más antigua y simple de las formas de lógica. Utilizando una representación primitiva del lenguaje, permite



representar y manipular aserciones sobre el mundo que nos rodea. La lógica proposicional permite el razonamiento, a través de un mecanismo que primero evalúa sentencias simples y luego sentencias complejas, formadas mediante el uso de conectivos proposicionales, por ejemplo Y (AND), O (OR). Este mecanismo determina la veracidad de una sentencia compleja, analizando los valores de veracidad asignados a las sentencias simples que la conforman [Borrego, 2002].

Una proposición es una sentencia simple que tiene un valor asociado ya sea de verdadero (V), o falso (F). Por ejemplo:

*Hoy es Viernes*

*Ayer llovió*

*Hace frío*

La lógica proposicional, permite la asignación de un valor verdadero o falso para la sentencia completa, no tiene facilidad para analizar las palabras individuales que componen la sentencia. Por este motivo, la representación de las sentencias del ejemplo, como proposiciones, sería:

*hoy\_es\_Viernes*

*ayer\_llovió*

*hace\_frío*

Las proposiciones pueden combinarse para expresar conceptos más complejos. Por ejemplo:

*hoy\_es\_Viernes y hace\_frío.*

A la proposición anterior dada como ejemplo, se la denomina **fórmula bien formada** (*well-formed formula*, **wff**). Una fórmula bien formada puede ser una proposición simple o compuesta que tiene sentido completo y cuyo valor de veracidad, puede ser determinado. La lógica proposicional proporciona un mecanismo para asignar valores de veracidad a la proposición compuesta, basado en los valores de

veracidad de las proposiciones simples y en la naturaleza de los conectores lógicos involucrados.

### **2.3.1.2 Representación mediante Lógica de Predicados.**

La principal debilidad de la lógica proposicional es su limitada habilidad para expresar conocimiento. Existen varias sentencias complejas que pierden mucho de su significado cuando se las representa en lógica proposicional. Por esto se desarrolló una forma lógica más general, capaz de representar todos los detalles expresados en las sentencias, esta es la lógica de predicados.

La lógica de predicados está basada en la idea que las sentencias realmente expresan relaciones entre objetos, así como también cualidades y atributos de tales objetos. Los objetos pueden ser personas, objetos físicos, o conceptos. Tales cualidades, relaciones o atributos, se denominan predicados. Los objetos se conocen como argumentos o términos del predicado.

Al igual que las proposiciones, los predicados tienen un valor de veracidad, pero a diferencia de las proposiciones, su valor de veracidad, depende de sus términos. Es decir, un predicado puede ser verdadero para un conjunto de términos, pero falso para otro [Torres, 2001].

Por ejemplo, el siguiente predicado es verdadero:

*color (yerba, verde)*

El mismo predicado, pero con diferentes argumentos, puede no ser verdadero:

*color (yerba, azul) o color (cielo, verde)*

Los predicados también pueden ser utilizados para asignar una cualidad abstracta a sus términos, o para representar acciones o relaciones de acción entre dos objetos. Por ejemplo:

<i>mortal(juan_carlos)</i>	<i>clima(martes, lluvioso)</i>	<i>ave(gaviota)</i>	<i>ama(roberto, vanessa)</i>	<i>lee(alex, novela)</i>
----------------------------	--------------------------------	---------------------	------------------------------	--------------------------

Al construir los predicados se asume que su veracidad está basada en su relación con el mundo real. Naturalmente, siendo prácticos, trataremos que los predicados que definimos estén de acuerdo con el mundo que conocemos, pero no es absolutamente necesario que así lo hagamos. En lógica de predicados el establecer como verdadero un predicado es suficiente para que así sea considerado [Cañas, 2005].

Demos el siguiente ejemplo, que indica que Ecuador está en Europa:

*parte\_de(ecuador, europa)*

Obviamente, esto no es verdadero en el mundo real, pero la lógica de predicados no tiene razón de saber geografía y si el predicado es dado como verdadero, entonces es considerado como lógicamente verdadero. Tales predicados, establecidos y asumidos como lógicamente verdaderos se denominan axiomas, y no requieren de justificación para establecer su verdad.

La lógica de predicados, se ocupa únicamente de métodos de argumentación sólidos. Tales argumentaciones se denominan Reglas de Inferencia. Si se da un conjunto de axiomas que son aceptados como verdaderos, las reglas de inferencia garantizan que sólo serán derivadas consecuencias verdaderas.

Tanto los conectivos lógicos, como los operadores dados anteriormente para la lógica proposicional, son igualmente válidos en lógica de predicados. De hecho, la lógica proposicional es un subconjunto de la lógica de predicados.

Cada uno de los argumentos en los ejemplos de predicados dados anteriormente, representan a un objeto específico. Tales argumentos se denominan constantes. Sin embargo, en la lógica de predicados se

pueden tener argumentos que en determinado momento pueden ser desconocidos. Estos son los argumentos tipo *variable*.

En el ejemplo: *color (yerba, X)*, la variable *X*, puede tomar el valor de *verde*, haciendo que el predicado sea verdadero; o puede tomar el valor de *azul*, dando lugar a que el predicado sea falso.

Las variables, también pueden ser cuantificadas. Los cuantificadores que típicamente se utilizan en lógica de predicados son:

- El cuantificador universal;  $\forall$  indica que la fórmula bien formada, dentro de su alcance, es verdadera para todos los valores posibles de la variable que es cuantificada. Por ejemplo:

$$X \dots \forall$$

Establece que "para todo *X*, es verdad que . . . "

- El cuantificador existencial;  $\exists$ , indica que la fórmula bien formada, dentro de su alcance, es verdadera para algún valor o valores dentro del dominio. Por ejemplo:

$$X \dots \exists$$

Establece que "existe un *X*, tal que . . . "

A continuación se dan algunos ejemplos de predicados cuantificados:

$$X, [\text{niño}(X) = \forall > \text{le\_gusta}(X, \text{helados})].$$

$$Y, [\text{mamífero}(Y) = \forall > \text{nace}(Y, \text{vivo})].$$

$$Z, [\text{cartero}(Z) \wedge \text{mordió}(\text{boby}, Z)].\exists$$

Desde el punto vista de representación, los cuantificadores son difíciles de usar. Por lo que es deseable reemplazarlos con alguna representación equivalente, más fácil de manipular.

El cuantificador existencial es más difícil de reemplazar. El cuantificador existencial garantiza la existencia de uno o más valores particulares (instancias) de la variable cuantificada, que hace a la cláusula verdadera.

### 2.3.1.3 Reglas de Producción.

Los sistemas basados en reglas son los más comúnmente utilizados. Su simplicidad y similitud con el razonamiento humano, han contribuido para su popularidad en diferentes dominios. Las reglas son un importante paradigma de representación del conocimiento.

Las reglas representan el conocimiento utilizando un formato **SI-ENTONCES (IF-THEN)**, es decir tienen 2 partes:

- La parte **SI (IF)**, es el antecedente, premisa, condición o situación; y
- La parte **ENTONCES (THEN)**, es el consecuente, conclusión, acción o respuesta.

Las reglas pueden ser utilizadas para expresar un amplio rango de asociaciones, por ejemplo:

**SI** está manejando un vehículo **Y** se aproxima una ambulancia, **ENTONCES** baje la velocidad **Y** hágase a un lado para permitir el paso de la ambulancia.

**SI** su temperatura corporal es de 39 °C, **ENTONCES** tiene fiebre.

**SI** el drenaje del lavabo está tapado **Y** la llave de agua está abierta, **ENTONCES** se puede inundar el piso.

**Inferencia Basada en Reglas:** Una declaración de que algo es verdadero o es un hecho conocido, es una afirmación.

El conjunto de afirmaciones se conoce a menudo con el nombre de memoria de trabajo o base de afirmaciones. De igual forma, al conjunto de reglas se lo denomina base de reglas.

Un sistema basado en reglas utiliza el modus ponens para manipular las afirmaciones y las reglas durante el proceso de inferencia. Mediante técnicas de búsqueda y procesos de unificación, los sistemas basados en reglas automatizan sus métodos de razonamiento y proporcionan una progresión lógica desde los datos iniciales, hasta las conclusiones deseadas. Esta progresión hace que se vayan conociendo nuevos hechos o descubriendo nuevas afirmaciones, a medida que va guiando hacia la solución del problema.

En consecuencia, 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 las realiza en forma progresiva, por lo que, se dice que el proceso de solución origina una cadena de inferencias.

Los sistemas basados en reglas difieren de la representación basada en lógica en las siguientes características principales:

- Son en general no-monotónicos, es decir hechos o afirmaciones derivadas, pueden ser retractados, en el momento en que dejen de ser verdaderos.
- Pueden aceptar incertidumbre en el proceso de razonamiento.

### **2.3.2.1 Redes Semánticas**

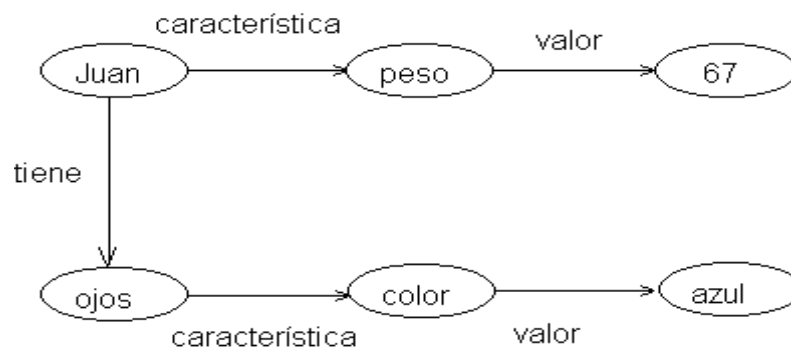
Las redes semánticas fueron desarrolladas por quienes trabajan en el área de la inteligencia artificial. Las estructuras básicas de este modelo consisten en nodos y arcos formando una red (un grafo). El objetivo de estas redes es la organización y representación del conocimiento general acerca del mundo.

El objetivo inicial para el desarrollo de las redes semánticas fue el entender el lenguaje natural, más que la clasificación de datos. Otra característica de las redes semánticas es que existen tantas como las necesidades que han tenido diferentes investigadores en diferentes proyectos.

Así, resulta difícil decidir a qué se le llama modelo de datos Red Semántica. Esto se debe a que se han tenido diferentes modelos de datos red semántica que son buenos al representar una realidad específica. Se puede decir entonces que cualquier grafo en el cual los nodos se conecten por medio de arcos se le puede llamar red semántica, siempre que nodos y arcos estén etiquetados.

Para tener semántica en un grafo, se necesita definir cuidadosamente el significado de nodos y arcos, y cómo son usados [Moreno, 2000].

Las primeras redes semánticas usaban diferentes nodos y arcos para representar las asociaciones presentes en la memoria humana. Estas primeras redes fueron poco uniformadas en su estructura, no distinguiendo adecuadamente entre diferentes tipos de nodos y arcos. Por ejemplo, objetos individuales (instancias) y clases de objetos (entidades) coexistían en la misma red semántica. No se tenía una clara diferencia entre los nodos que denotaban instancias y los que denotaban clases, por ejemplo, las siguientes redes semánticas.



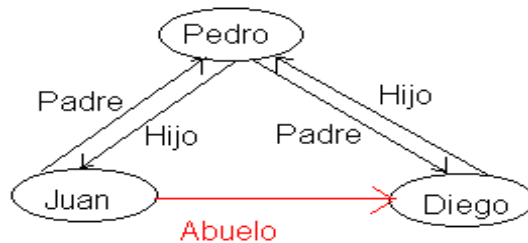
**Figura II.1 Redes Semánticas Básicas**

### **Características.**

**a.** Diferencian entre tipos de objetos de las instancias. Así, se llama clasificación al proceso de ir de instancias de objetos a tipos de objetos.

**b.** Se introduce el concepto de distancia semántica, cantidad de arcos que separan un nodo de otro. En otros modelos de datos esta distancia sólo tiene implicancias en la performance, y generalmente no se considera ni tiene ninguna connotación semántica [Jiménez, 2002].

En las redes semánticas la distancia puede ser importante, y es usada para localizar objetos poco o muy relacionados, dependiendo de la distancia. En algunos casos se puede disminuir la distancia agregando arcos con ese propósito.



**Figura II.2 Distancia Semántica**

c. En las redes semánticas también se tiene la idea de partición: es el contexto de una red, en el sentido de tener una subred, así para una tarea o trabajo específico sólo una parte de la red está disponible. Esta facilidad resulta útil en el momento de realizar búsquedas, ya que se limita el espacio de búsqueda.

d. También se tiene la jerarquía de tipo (u objeto). Los tipos de jerarquías que se tienen en una red semántica son PARTE-DE y ES-UN. La existencia de una jerarquía implica que se permite la herencia donde un objeto que pertenece a una clase hereda todas las propiedades de la clase.

La herencia no se refiere a la herencia de atributos y sus valores, sino que también se heredan los tipos de relaciones permitidas para esa clase. Ejemplo: Si un empleado es una persona, y la relación "casado con" es válida para persona, entonces también es válida para Ingeniero, Abogado y Secretaria.

### **2.3.2.2 Estructuras Frames (Marcos).**

Es una forma de organizar la información referente a un objeto, donde se describen las propiedades del objeto en forma similar a un registro de una



base de datos, donde se reservan los campos, para cada propiedad [Pacheco, 2007].

Un marco es una estructura de datos compleja que contiene información acerca de un objeto. La información almacenada en el marco se distribuye en diferentes campos llamados ranuras o aspectos y cada una de ellas contiene la información sobre un atributo del objeto o un apuntador a otro marco.

Ejemplos:

1- Marco: Empleado

Nombre

Edad (entre 16 y 65)

Piernas (2 por defecto)

Ocupación

2- Marco: Hombre de familia

Nombre

Edad (entre 16 y 120)

Estado civil (genérico: casado)

Cónyuge (Apuntador al marco Mujer de familia)

Se denomina ejemplificación de un marco a uno con sus ranuras llenas. Podemos ver entonces al marco como una clase de entidades y a una ejemplificación de él como una entidad particular.

Ejemplos de ejemplificaciones de los marcos anteriores son:

1- Marco: Empleado

Nombre: Jesús López

Edad: 32

Piernas: 2

Ocupación: Programador

2- Marco: Hombre de familia

Nombre: Jesús López

Edad: 32

Estado civil: casado

Cónyuge: (Apuntador a la ejemplificación para Mujer de familia de Mariela Pérez)

A cada ranura se le puede asociar varios tipos de información, llamados facetas de la ranura, entre las cuales están:

**Valor:** almacena el valor para la ranura. Ejemplo: la ranura Nombre de los marcos Empleado y Hombre de familia.

**Valor por defecto:** contiene un valor inicial o valores usados comúnmente para la faceta Valor. Ejemplo: la mayoría de las personas empleadas poseen dos piernas y salvo se conozca otra cosa, se puede pensar como normal tal afirmación. Usualmente se utilizan valores por defecto cuando no se dispone de conocimiento de lo contrario. En ocasiones el valor por defecto se puede usar como el valor inicial de la faceta Valor.

**Genérico:** Es una propiedad asociada a una clase de entidades, de tal forma que cada entidad de la clase posee esa propiedad. Ejemplo: Casado en el marco Hombre de familia.

**Restricciones:** contiene un conjunto de expresiones lógicas que tienen que ser verdaderas para el valor almacenado en la faceta Valor. Ejemplo: la ranura Edad en el marco Empleado. Las restricciones a analizar pueden ser más complejas y pueden referirse a valores de otras ranuras del marco actual o incluso de otro marco. Ejemplo: supongamos que al marco Empleado le agregamos una ranura Sexo y que hacemos las consideraciones laborales en nuestro país. La condición de la edad va a depender, entonces, de si es hombre o mujer el empleado.

**Procedimiento:** contiene un procedimiento para calcular el valor que debe ser almacenado en la faceta Valor.

**Demon:** contiene un procedimiento que se ejecuta automáticamente cuando se cambia el valor almacenado en la faceta Valor. Es importante

pues permite mantener la consistencia interna de la información.

Ejemplos:

- Un demon asociado a la ranura Edad del marco Empleado que se activará si se trata de rellenar con un valor fuera del rango de 16 a 65. El demon puede pedir explicación al usuario de la causa de la violación y actuar según la explicación dada.

- Al rellenarse la ranura de Estado civil en Hombre de familia con el valor casado, se puede activar un demon que ejemplifique otro marco de Mujer de familia para el cónyuge.

Tanto los procedimientos como los demons son escritos, normalmente, como programas en lenguajes procedurales de alto nivel.

**Explicación:** almacena documentación sobre la ranura.

Hemos mencionado 7 facetas de una ranura. No necesariamente, cada ranura debe tenerlas todas. Eso depende de cada problema en particular.

En general, se puede plantear que la información almacenada usando como F.R.C. a los marcos tiene cuatro niveles de detalle: el marco, las ranuras del marco, las facetas de cada ranura y el valor almacenado en cada faceta.

### **2.3.2.3 Representación orientada a objetos.**

La representación de conocimiento orientada a objetos suministra una alternativa, pero cierra relativamente la metodología para representar conocimiento que comparten un conjunto de estructuras como las redes semánticas y los marcos. Esta representación es vista en términos de un conjunto de objetos, los cuales son capaces de seguir ciertos tipos de conductas. Cada objeto se sitúa jerárquicamente en una red y puede acceder a propiedades e información de los objetos de un nivel superior.

Los argumentos hechos para los objetos como estructuras de conocimiento se relacionan con el estilo de descripción inicialmente

usado, más que las propiedades fundamentales de objetos. El estilo de descripción de objetos es un proceso de comparación, así que un nuevo objeto se describe para determinar la forma en que sea similar o diferente a los objetos ya conocidos.

En el paradigma orientado a objetos, estos se comunican unos con otros por envío y recibo de mensajes.

- Un objeto, muy similar a los marcos recibe mensajes de verificación de conocimiento y decide que acción tomar.
- La descripción para comparar muchas de las similitudes es mostrar una especialización en casos usando los marcos.
- Los objetos usan atributos que están especificados en ranuras; se organizan jerárquicamente de manera que los de menor nivel comparten propiedades con los de mayor nivel (herencia).
- Los objetos contienen métodos, mientras los marcos tienen procedimientos asociados.

## **2.4 Conclusiones.**

- Estudiamos los detalles, clasificación y tipos de representación de conocimiento, aprendimos que es necesario realizar la selección del tipo más adecuado de representación para cada necesidad dependiendo de los formalismos y el lenguaje utilizado para su implementación.
- En el Capítulo analizamos las formas de representar el conocimiento mediante lógica proposicional, lógica de predicados, redes semánticas, marcos y orientado a objetos. Además la importancia que tiene la representación del conocimiento para las aplicaciones de inteligencia artificial y los tipos procedimental, declarativo y heurístico.

## **CAPÍTULO III. Técnicas de Inteligencia Artificial.**

### **3.1. Introducción**

En este Capítulo se pretende comprender el funcionamiento básico, características y ventajas de las técnicas de Inteligencia Artificial. Estudiar sus principales técnicas como son: redes neuronales, lógica difusa, algoritmos genéticos, los cuales mediante su utilización han ayudado enormemente a la mejora de procesos, la calidad, control y la producción en serie, también se han utilizado para la toma de decisiones más eficaz o en la ayuda de labores o tareas que exijan gran demanda de tiempo o representen un alto grado de peligrosidad para el ser humano.

Al final de este Capítulo se realiza también un resumen de las formas de solución de problemas mediante los distintos tipos y clasificación de búsquedas en Inteligencia Artificial, como búsqueda Ciega, Búsqueda en Profundidad, Búsqueda Heurística entre las principales pues es importante conocer la forma en que una determinada técnica llega a la solución de un problema.

### **3.2 Qué es una técnica de Inteligencia Artificial (IA)**

La Inteligencia Artificial engloba un conjunto de técnicas que tienen en común la robustez en el manejo de la información imprecisa que existe en los problemas relacionados en el mundo real. Estas técnicas son:

- Redes Neuronales.
- Algoritmos Genéticos.
- Sistemas de Lógica difusa.

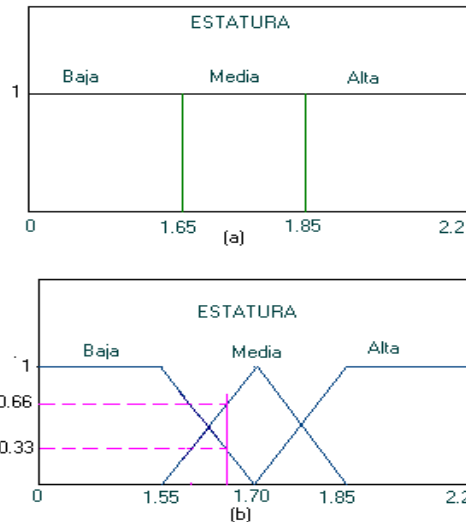
Pero se han venido destacando otros paradigmas como son los agentes de decisión inteligente y autómatas programables, con respecto a estos últimos se suelen emplear en gran medida en procesos industriales de acuerdo a necesidades a satisfacer como, espacio reducido, procesos de producción periódicamente cambiantes, procesos secuenciales, maquinaria de procesos variables, etc. También estas técnicas pueden ser agrupadas para aprovechar sus ventajas individuales.

A continuación se explicaran brevemente las técnicas: lógica difusa, redes neuronales y algoritmos genéticos.

### **3.2.1 Lógica Difusa.**

La lógica clásica, o lógica bivaluada, no resulta adecuada cuando de describir el razonamiento humano se trata, ya que solo “conoce” dos valores, verdad (1) y falsedad (0), mientras que en la vida real existen hechos que no se pueden definir como totalmente verdaderos o totalmente falsos sino que tienen un grado de verdad o falsedad, que puede variar de 0 a 1. Un ejemplo sencillo se puede apreciar cuando queremos saber si un vaso está lleno de agua, pero al observarlo notamos que éste no está totalmente lleno. Nuestro sentido común no asigna inmediatamente el valor de 0 (falsedad) a nuestra inquietud sino que razona aceptando que está “algo lleno”, es decir tiene una alta posibilidad de ser catalogado como lleno, lo cual se puede representar con un valor más cercano a 1 que a 0.

La figura III.1a. Muestra una posible representación de la variable lingüística estatura mediante la lógica clásica bivaluada y la figura III.1b muestra una posible representación de esa misma variable lingüística mediante la lógica difusa. Por cada variable lingüística se han establecido tres valores lingüísticos, estatura baja, estatura media y estatura alta.



**Figura III.1. Variable lingüística estatura representada mediante:**

**(a) Lógica clásica y (b) lógica difusa.**

Se puede apreciar que en la lógica clásica, una persona de 1.65 es considerada de estatura baja, mientras que otra con un (1) milímetro más de estatura, es decir, 1.651 es considerada de estatura media. En la lógica difusa, esa misma persona tiene una posibilidad de ser de estatura baja de 0.33 y de ser de estatura media de 0.66, aproximadamente, es decir tiene más posibilidad de ser de estatura media que de estatura baja, pero tiene pertenencia a dos conjuntos diferentes. De esta manera, la lógica difusa permite resolver problemas de la vida real con alto nivel de incertidumbre, que difícilmente pueden ser resueltos por la lógica clásica.

La lógica difusa se desarrolló a partir de 1965 con la aparición de un artículo sobre la teoría de los conjuntos difusos, establecida por el catedrático de la Universidad de Berkeley, California, Lofti A. Zadeh, quien hoy es considerado el padre de la lógica difusa [Carbonell,2002].

**Tipos de Sistemas difusos:** Existen dos principales: los controladores difusos de tipo Mamdani y Sugeno, los cuales han sido utilizados exitosamente en una gran variedad de aplicaciones en la comunidad del control difuso. Aunque, el objetivo del controlador difuso Mamdani es el de representar a un exitoso operador humano, el controlador difuso de tipo Sugeno sugiere ser más eficiente en cálculos y en métodos de adaptación [Gómez, 2005].

### **3.2.1.1 Características.**

- Se basa en palabras y no en números, las verdades de los valores son expresados lingüísticamente. Por ejemplo: caliente, muy frío, verdad, lejano, cercano, rápido, lento, medio, etc.
- Ésta genera algunos modificadores del predicado como por ejemplo: mucho, más o menos, poco, suficientemente, medio, etc.
- También procesa un sistema amplio de cuantificadores, como por ejemplo: pocos, varios, alrededor, generalmente.
- Hace uso las probabilidades lingüísticas, como por ejemplo: probable, improbable, que se interpretan como números borrosos y son manipuladas por su aritmética.
- Maneja todos los valores entre 0 y 1, tomando éstos como límite solamente.
- Usa una representación de conocimiento explícito.
- Realiza verificación y optimización de manera fácil y eficiente.
- No se puede entrenar, esto es que sea capaz de obtener nuevos conocimiento.

### **3.2.1.2 Etapas de la Lógica Difusa.**

**1. Fusificación (Fuzzification).** Las funciones de pertenencia definidas para las variables de entrada se aplican a sus valores actuales correspondientes, para poder determinar el grado de verdad para cada regla de la premisa.

**2. Inferencia Lógica.** El valor de verdad para la premisa de cada regla se calcula, y aplica a la parte de conclusiones de cada regla. Este resultado se asigna a un subconjunto difuso para ser asignado a cada variable de salida para cada regla.

**3. Difusificación (Defuzzification).** La cual es usada cuando se desea convertir la salida difusa en un valor puntual numérico. Existen muchos métodos de dedifusificación (al menos 30).



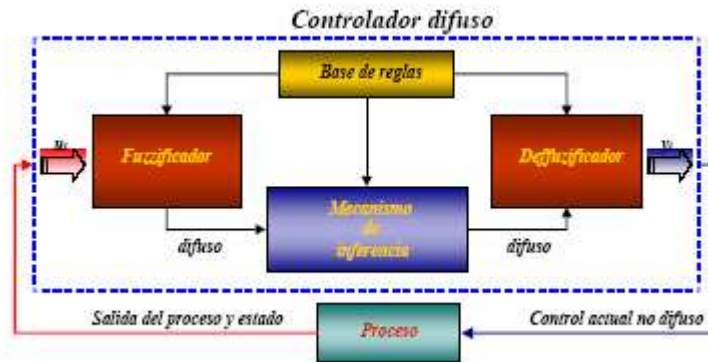


Figura III.2. Esquema general del control difuso.

### 3.2.1.3 Aplicaciones.

Actualmente existen en nuestros países todo tipo de instrumentos, máquinas y procedimientos controlados borrosamente, adaptándose "inteligentemente" a cada situación particular: acondicionadores de aire, frigoríficos, lavadoras / secadoras, aspiradoras, hornos microondas, mantas eléctricas, ventiladores, autoenfoques fotográficos, estabilizadores de imágenes en grabadoras de vídeo, transmisiones de automóviles, suspensiones activas, controles de ascensores, procesadores de arroz, traductores dispensadores de anticongelantes para los aviones en los aeropuertos, sistemas de toma de decisiones industriales o económicas, algunos correctores de voz para sugerir una lista de probables palabras a reemplazar en una mal dicha, etcétera.

Un ejemplo, en las lavadoras "difusas" tiene más de 400 ciclos preprogramados; a pesar de su complejidad tecnológica resultan más fáciles de operar que las lavadoras tradicionales. El usuario solo pone en marcha la lavadora, el resto queda en manos del control difuso. El evalúa automáticamente el material, el volumen, la suciedad de la ropa, elige el ciclo óptimo de lavado, así como el caudal de agua que ha de emplear.

En las actividades domésticas cotidianas se inventó un sistema de ventilación que usa el control difuso para conmutar un ventilador según

los conocimientos de cantidad de polvo, olores, temperatura y humedad ambiente [Universidad de las Américas, 1999].

El baño difuso, por ejemplo tiene un controlador que mantiene el agua a la temperatura ideal del usuario, ni muy fría ni muy caliente.

Una de las aplicaciones más importantes de la teoría difusa se aplicaron en Sendai (Japón) donde se puso en servicio un metro controlado mediante lógica difusa, este tenía la peculiaridad de poseer controladores que hacían las frenadas y la aceleraciones mucho más suaves facilitando así la conducción.

A partir de los 90 se incluyó también en los ascensores consiguiendo así reducir el tiempo de espera y la seguridad de los mismos. También se ha comenzado a implementar en los controles de inyección electrónica del carburante y en los sistemas de guiado automático de coches, haciendo los controles complejos más eficientes y fáciles de utilizar

### **3.2.2 Redes Neuronales.**

Las Redes Neuronales Artificiales (RNA) tratan de modelar la estructura y la forma de trabajar de las células nerviosas o neuronas. Las neuronas artificiales son interconectadas unas a otras y son distribuidas en capas de tal forma que emulan en forma simple la estructura neuronal de un cerebro. Cada modelo de neurona es capaz de realizar algún tipo de procesamiento a partir de estímulos de entrada y ofrecer una respuesta, por lo que las RNA en conjunto funcionan como redes de computación paralelas y distribuidas similares a los sistemas cerebrales biológicos [Fuente, 2000].

Sin embargo, a diferencia de las computadoras convencionales, las cuales son programadas para realizar tareas específicas, las redes neuronales artificiales, tal como los sistemas cerebrales biológicos, deben ser entrenadas.

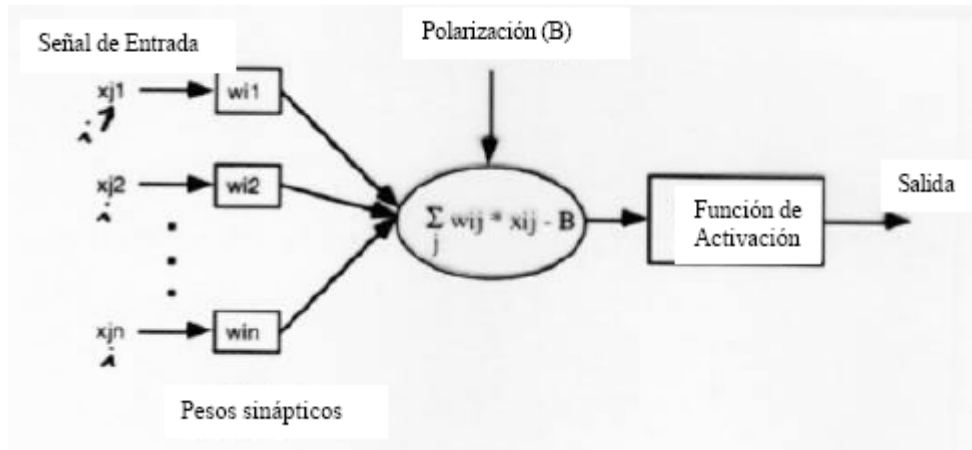
**Diseño y Programación de una Red Neuronal Artificial:** Con un paradigma convencional de programación informática, el objetivo del programador es modelar matemáticamente (con distintos grados de formalismo) el problema en cuestión y formular una solución (programa) mediante un algoritmo que tenga una serie de propiedades que permitan resolver dicho problema.

En contraposición, la aproximación basada en las RNA parte de un conjunto de datos de entrada suficientemente significativo y el objetivo es conseguir que la red aprenda automáticamente las propiedades deseadas. En este sentido, el diseño de la red tiene menos que ver con cuestiones como los flujos de datos y la detección de condiciones, y más que ver con cuestiones tales como la selección del modelo de red, la de las variables a incorporar y el reprocesamiento de la información que formará el conjunto de entrenamiento. Asimismo, el proceso por el que los parámetros de la red se adecuan a la resolución de cada problema no se denomina genéricamente programación sino que se suele denominar entrenamiento.

Por ejemplo, en una red que se va a aplicar al diagnóstico de imágenes médicas; durante la fase de entrenamiento el sistema recibe imágenes de tejidos que se sabe son cancerígenos y tejidos que se sabe son sanos, así como las respectivas clasificaciones de dichas imágenes. Si el entrenamiento es el adecuado, una vez concluido, el sistema podrá recibir imágenes de tejidos no clasificados y obtener su clasificación sano/no sano con un buen grado de seguridad. Las variables de entrada pueden ser desde los puntos individuales de cada imagen hasta un vector de características de las mismas que se puedan incorporar al sistema (por ejemplo, procedencia anatómica del tejido de la imagen o la edad del paciente al que se le extrajo la muestra).

### 3.2.2.1 Estructura RNA.

Están compuestas por unidades estructurales también conocidas como neuronas, conectadas entre sí. El modelo básico de una neurona es el siguiente:



**Figura III.3 Estructura Red Neuronal Artificial**

Típicamente las redes neuronales poseen una capa de entrada (input layer), una capa de salida (output layer), y una o varias capas ocultas intermedias (hidden layers) [Vega, 2000].

La mayoría de los científicos coinciden en que una RNA es muy diferente en términos de estructura de un cerebro animal. Al igual que el cerebro una RNA se compone de un conjunto masivamente paralelo de unidades de proceso muy simples y es en las conexiones entre estas unidades donde reside la inteligencia de la red. Sin embargo, en términos de escala, un cerebro es muchísimo mayor que cualquier RNA creada hasta la actualidad, y las neuronas artificiales también son más simples que su contrapartida animal.

Biológicamente, un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De la misma manera, las RNA tienen un gran número de procesadores virtuales interconectados que de forma simplificada simulan la funcionalidad de las neuronas biológicas. En esta simulación, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de

pesos, que son ajustados durante la fase de aprendizaje. En una RNA entrenada, el conjunto de los pesos determina el conocimiento de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada.

Por otra parte, en una RNA, además de los pesos y las conexiones, cada neurona tiene asociada una función matemática denominada función de transferencia. Dicha función genera la señal de salida de la neurona a partir de las señales de entrada. La entrada de la función es la suma de todas las señales de entrada por el peso asociado a la conexión de entrada de la señal. Algunos ejemplos de funciones de transferencia son la función escalón, la lineal o mixta, la sigmoidea y la función gaussiana.

La topología de una red neuronal estará determinada por el número de neuronas que esta tenga, por la forma en éstas estén ordenadas (en capas) y por la naturaleza de las conexiones presentes.

### **3.2.2.2 Características y Ventajas.**

Las Redes de Neuronas Artificiales (RNA) tienen muchas ventajas debido a que están basadas en la estructura del sistema nervioso, principalmente el cerebro [Daza, 1997].

- **Aprendizaje:** Las RNA tienen la habilidad de aprender mediante una etapa que se llama etapa de aprendizaje. Esta consiste en proporcionar a la RNA datos como entrada a su vez que se le indica cuál es la salida (respuesta) esperada.
- **Auto organización:** Una RNA crea su propia representación de la información en su interior, descargando al usuario de esto.
- **Tolerancia a fallos.** Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo aceptablemente aún si se daña parcialmente.
- **Flexibilidad:** Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (ej. si la información de entrada es la imagen de un objeto,

la respuesta correspondiente no sufre cambios si la imagen cambia un poco su brillo o el objeto cambia ligeramente).

- Tiempo real: La estructura de una RNA es paralela, por lo cual si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real.
- Capacidad de Asociación (Generalización).
- Seguridad ante caídas del sistema.
- Procesamiento masivamente paralelo.
- Representación y Procesamiento distribuido de la información.

### 3.2.2.3 Modelos.

Existe una serie de modelos que aparecen en la mayoría de estudios académicos y la bibliografía especializada.

Diferentes modelos de redes son utilizados para resolver diferentes tipos de problemas. En la siguiente tabla puede observarse algunos de los modelos más utilizados y su campo de aplicación.

Modelo	Campo de Aplicación
Red de Hopfield	Optimización y Asociación de Patrones
Multilayer Perceptron (*Back-Propagation*)	Aproximación de Funciones, Clasificación, Pronóstico y Control
SOM – Self Organizing Map (Red de Kohonen)	Clustering y Clasificación
LVQ - Learning Vector Quantization	Clasificación
ART - Adaptive Resonance Theory	Clustering y Clasificación
Radial Basis Function	Aproximización de Funciones, Clasificación, Pronóstico y Control
Counterpropagation	Clasificación y Aproximización de Funciones
ADALINE-MADALINE	Sistemas de Comunicación y Control

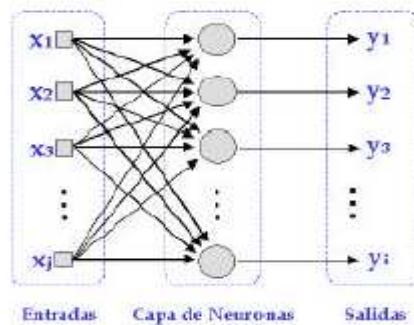
**Tabla III.1. Modelos y Campos de Aplicación RNA**

### 3.2.2.4 Arquitectura de Redes Neuronales Artificiales.

Diferentes tipos de interconexión implican diferentes comportamientos de la red. El flujo de la información puede ser unidireccional desde la capa de entrada hacia la capa de salida (redes feedforward) o bidireccional con retroalimentación (redes recurrentes).

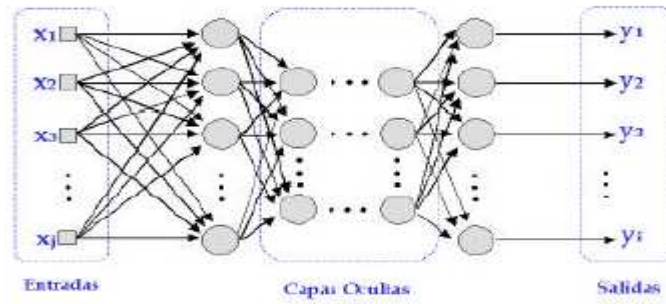
Por ejemplo, las redes que tienen flujo de datos unidireccional (feedforward) son estáticas, es decir, producen sólo un grupo de valores de salida en lugar de una secuencia de valores de salida para una entrada dada, además sus salidas no dependen de los valores anteriores de la red. Por otro lado las redes neuronales recurrentes o realimentadas son sistemas dinámicos. Según la arquitectura de la conexión las redes neuronales se pueden clasificar, entre otras, como: Red Neuronal Monocapa, Red Neuronal Multicapa y Red Neuronal Realimentada.

**Red Neuronal Monocapa:** Las redes monocapa están formadas sólo por una capa de neuronas, y suelen utilizarse frecuentemente en tareas relacionadas con la regeneración de información incompleta o distorsionada que se presenta a la red.



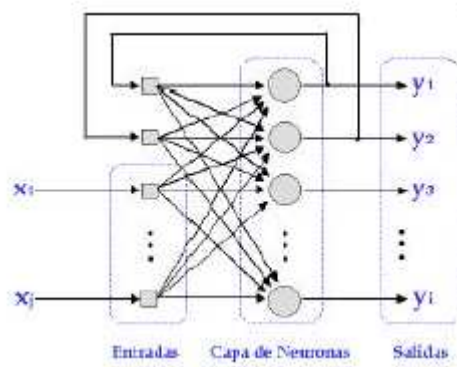
**Figura III.4. Red Neuronal Monocapa**

**Red Neuronal Multicapa:** Son las estructuras más comunes; además de la capas de entrada y salida, poseen un número de capas intermedias u ocultas que mejoran su desempeño.



**Figura III.5. Red Neuronal Multicapa**

**Red Neuronal Recurrente:** Se caracteriza porque sus salidas pueden ser utilizadas como entradas. La estabilidad de la red es un importante factor a considerar en este tipo de arquitectura [Fuente, 2000].



**Figura III.6. Red Neuronal Realimentada**

**Aprendizaje:** Una de las principales características de las redes neuronales es su capacidad de aprendizaje, esto se logra a través de la repetitiva presentación de ejemplos, lo cual permite adaptar los pesos sinápticos de las neuronas.

Otra clasificación que se suele hacer es en función del tipo de aprendizaje de que es capaz (si necesita o no un conjunto de entrenamiento supervisado). Para cada tipo de aprendizaje encontramos varios modelos propuestos por diferentes autores:

- Aprendizaje supervisado: necesitan un conjunto de datos de entrada previamente clasificado o cuya respuesta objetivo se conoce.



Ejemplos de este tipo de redes son: el perceptrón simple, la red Adaline, el perceptrón multicapa y la memoria asociativa bidireccional.

- Aprendizaje no supervisado o auto organizado: no necesitan de tal conjunto previo. Ejemplos de este tipo de redes son: las memorias asociativas, las redes de Hopfield, la máquina de Bolzman y la máquina de Cauchy, las redes de aprendizaje competitivo, las redes de Kohonen o mapas auto organizados y las redes de resonancia adaptativa (ART) .
- Redes híbridas: son un enfoque mixto en el que se utiliza una función de mejora para facilitar la convergencia. Un ejemplo de este último tipo son las redes de base radial.

### **3.2.2.5 Aplicaciones.**

Las características de las RNA las hacen bastante apropiadas para aplicaciones en las que no se dispone a priori de un modelo identificable que pueda ser programado, pero se dispone de un conjunto básico de ejemplos de entrada (previamente clasificados o no).

Esto incluye problemas de clasificación y reconocimiento de patrones de voz, imágenes, señales, etc. Asimismo se han utilizado para encontrar patrones de fraude, hacer predicciones en el mercado financiero, hacer predicciones de tiempo atmosférico, etc.

También se pueden utilizar cuando no existen modelos matemáticos precisos o algoritmos con complejidad razonable.

Otro tipo especial de redes neuronales artificiales se ha aplicado en conjunción con los algoritmos genéticos (AG) para crear controladores para robots. En este tipo de aplicación el genoma del AG lo constituyen los parámetros de la red (topología, algoritmo de aprendizaje, funciones de activación, etc.) y la adecuación de la red viene dada por la adecuación del comportamiento exhibido por el robot controlado (normalmente una simulación de dicho comportamiento).

En resumen las redes neuronales artificiales (RNA) son modelos computacionales que tratan de replicar, de manera simplificada, el complejo funcionamiento del cerebro humano utilizando métodos o modelos matemáticos muy sofisticados. Su capacidad de aprendizaje a través de ensayos repetidos, las ha hecho muy populares en una amplia variedad de aplicaciones en todas las ciencias. En este sentido las redes neuronales resultan una herramienta fundamental ya que proponen una estrategia de resolución de problemas basada en la adaptación al entorno de información.

### **3.2.3 Algoritmos Genéticos (AG).**

Los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

También los algoritmos genéticos son métodos de optimización: dada una función objetivo, el AG lleva a cabo una búsqueda en el espacio de soluciones intentando encontrar la solución que minimiza dicha función objetivo. Por lo tanto, los AG están indicados para resolver todo tipo de problemas que puedan ser expresados como un problema de optimización.

Básicamente, los algoritmos genéticos (AG) funcionan como sigue: dada una población de soluciones, y en base al valor de la función objetivo para cada una de los individuos (soluciones) de esa población, se seleccionan los mejores individuos (los que minimizan la función objetivo) y se combinan para generar otros nuevos. Este proceso se repite cíclicamente [García, 2001].

Como se puede ver, el proceso es similar al que se da en la naturaleza: una serie de individuos compiten por su supervivencia; los mejor adaptados al medio (los que optimizan la función objetivo) sobreviven y tienen más posibilidades de aparearse, transmitiendo así parte de su material genético "mejor adaptado" a las generaciones siguientes. De esta manera, generación tras generación, la especie consigue una mayor y mejor adaptación al medio en el que vive.

**Algo de Historia:** Desde finales de los años 50 y principios de los 60 se desarrollaron trabajos por algunos biólogos para simular sistemas genéticos en una computadora. A pesar de estos trabajos, se reconoce al profesor John Holland de la Universidad de Michigan como el creador de los AG con su trabajo sobre teoría de los sistemas adaptativos, en el año 62. Es Holland el primero que adaptó la idea de la genética a sistemas artificiales. Su objetivo era lograr que las computadoras aprendieran por sí mismas. A la técnica que inventó Holland se le llamó originalmente "planes reproductivos", pero se hizo popular bajo el nombre "algoritmo genético" tras la publicación de su libro en 1975.

**Terminología Usada:** En el trabajo con AG se maneja una serie de términos "importados" de la genética natural. No siempre es adecuada la analogía, pero estos son comúnmente aceptados:

Población	Conjunto de individuos o cromosomas. Equivale a una muestra aleatoria del espacio de solución o un conjunto de soluciones alternativas.
Cromosoma	Un cromosoma es un portador de la información genética que transmite cada uno de sus genes. Una posible solución.
Gen	Cada uno de los rasgos o características que conforman el cromosoma. También se les llama parámetros o aspectos. Cada gen equivale a una variable del problema.
Genotipo	En biología se le llama al "paquete" genético total en su forma interna. En la terminología de AG será la información genética de todo el cromosoma en forma codificada.

Fenotipo	Se le llama en genética al paquete genético tal y como interactúa con el medio exterior. En los AG artificiales serían los aspectos del cromosoma decodificados.
Locus	Es la posición de un gen en el cromosoma
Alelo	Es el valor asociado a un gen

**Tabla III.2 Términos de Algoritmos Genéticos**

### **3.2.3.1 Componentes de un AG.**

Los AG trabajan a partir de una población inicial de estructuras artificiales que van modificando repetidamente a través de la aplicación de los siguientes operadores genéticos:

- Operador de Selección o Darwiniano
- Operador de Cruzamiento o Mendeliano
- Operador de Mutación.

Para utilizar los AG es necesario encontrar una posible estructura para representar las soluciones. Pensando este asunto como el problema de buscar en un espacio de estados, una instancia de esta estructura representa un punto o un estado en el espacio de búsqueda de todas las posibles soluciones. Así, una estructura de datos en el AG consistirá en uno o más cromosomas (frecuentemente uno), el cual se representa comúnmente como una cadena de bits (existen otras representaciones) [Rodríguez, 2004].

Cada cromosoma (cadena) es una concatenación de un número de subcomponentes llamados genes. La posición de un gen en el cromosoma se conoce como locus y sus valores como alelos. En la representación como cadena de bits, un gen es un bit o una cadena de bits, un locus es su posición en la cadena y un alelo es su valor (0 ó 1 si es un bit).

Los AG realizan una maximización por defecto, para los problemas de

minimización los valores de la función objetivo pueden ser negados y trasladados con vistas a tomar valores positivos para producir así la adaptabilidad.

### 3.2.3.2 Funcionamiento de un AG.

El modo de trabajo de un AG puede resumirse en el siguiente esquema el funcionamiento de un AG simple. El mecanismo de un AG simple es:

- El AG simple genera aleatoriamente una población de n estructuras (cadenas, cromosomas o individuos).
- Sobre la población actúan los operadores transformando la población. Una vez completada la acción de los tres operadores se dice que ha transcurrido un ciclo generacional.
- Luego se repite el paso anterior mientras no se garantice el criterio de parada del AG.

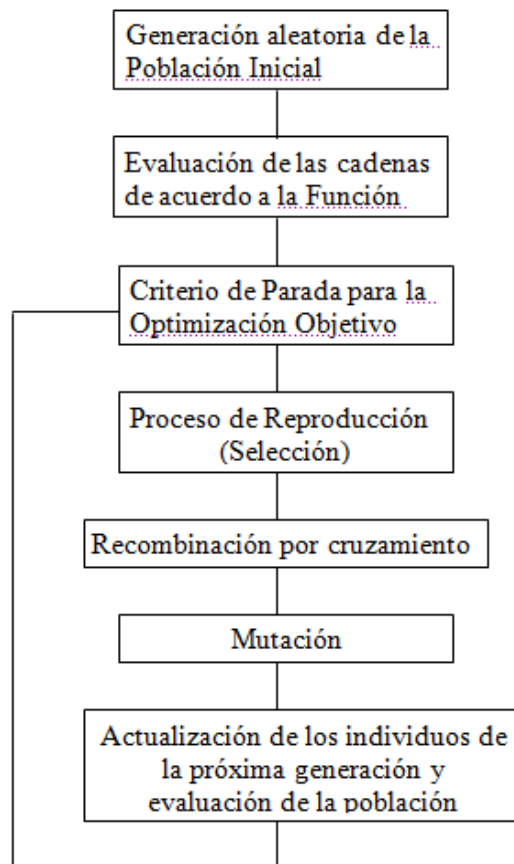


Figura III.7. Diagrama funcional de un algoritmo genético.

- El operador de selección o Darwiniano realiza la selección de las cadenas de acuerdo a su adaptabilidad para el posterior apareamiento.
- El operador de cruzamiento o Mendeliano realiza la recombinación del material genético de dos cadenas padres.
- El operador de Mutación al estilo del operador natural realiza la mutación de un gen dentro de un cromosoma o cadena a sus diferentes formas alelomorfos.
- Para cada uno de estos operadores está asociado el uso de probabilidades y la generación de números aleatorios.
- El AG ejecuta para un número fijo de generaciones o hasta que se satisface algún criterio de parada.

Desde el punto de vista de la comparación de los AG con otros métodos de búsqueda se pueden enmarcar sus diferencias en cuatro aspectos:

- 1- Trabajan con una codificación de los parámetros y no con los parámetros mismos.
- 2- Buscan a partir de una población de puntos y no de un punto simple.
- 3- Usan directamente la función objetivo y no la derivada u otro conocimiento auxiliar.
- 4- Usan reglas de transición probabilísticas y no determinísticas.

### **¿Por qué utilizar AG y no otros métodos conocidos?**

Las razones que se pueden argumentar son varias. Los métodos conocidos son buenos mientras el problema no es muy complejo. Los AG permiten la solución eficiente de funciones extremadamente complejas.

Las potencialidades de los AG se pueden resumir por sus habilidades para resolver una variedad de problemas muy difíciles:

- Trabajar sin conocimiento previo de la función a optimizar.
- Optimizar funciones “ruidosas”.
- Trabajar sin información secundaria como gradientes.

La mayoría de los especialistas en este tema coinciden en que los AG pueden resolver las dificultades representadas en los problemas de la vida real que a veces son insolubles por otros métodos.

Para [Rodríguez, 2004] el tema central de la investigación en AG consiste en la robustez: el balance entre la eficacia y la eficiencia necesaria para sobrevivir en muchos ambientes diferentes.

### **3.2.3.3 Aplicaciones tradicionales de AG.**

Los algoritmos genéticos se utilizan en sistemas de búsqueda y optimización, este es el campo en que más aplicaciones se reportan, habiéndose realizado incluso trabajos con funciones muy complejas. También se ha trabajado en la obtención de soluciones a ecuaciones no lineales [Costa, 2002].

Desde aplicaciones evidentes, como la biología o la medicina, hasta otros campos como la industria (clasificación de piezas en cadenas de montaje). Los algoritmos genéticos poseen un importante papel en este ámbito.

### **3.3 Búsqueda en Inteligencia Artificial.**

Los investigadores pioneros en Inteligencia Artificial tuvieron como su primer objetivo la solución de problemas que fueron difíciles de resolver mediante las técnicas computacionales existentes. Como se dijo antes, estos problemas generalmente no tienen solución algorítmica conocida o esta es tan compleja que no tiene una implementación práctica computacional.

La respuesta fue desarrollar nuevas técnicas de solución de problemas, similares a las humanas, una de las más importantes fue la búsqueda.

La búsqueda en la I.A. difiere de la búsqueda convencional sobre estructuras de datos esencialmente en que se busca en un espacio problema, no en una pieza de dato particular. Se busca un camino que conecte la descripción inicial del problema con una descripción del estado

deseado para el problema, es decir, el problema resuelto. Este camino representa los pasos de solución del problema.

El proceso de buscar una solución a un problema produce un espacio solución, o sea, la parte del espacio problema que se examina realmente. A diferencia de las estructuras de datos que están predefinidas y ya existen cuando comienza la búsqueda, los espacios problema son generalmente definidos proceduralmente, es decir, el espacio problema es creado a medida que es explorado. Se usan procedimientos para definir los siguientes estados posibles en el espacio a través de los cuales la búsqueda puede continuar desde el estado actual. Solamente los caminos explorados tienen que estar definidos explícitamente.

Hay diferentes alternativas para realizar la búsqueda. Desde un punto de vista podemos apreciar tres alternativas: aleatoria, a ciegas y dirigida. Con el siguiente ejemplo se ilustran estos. Supóngase que está en París sin un mapa y no habla francés ¿Cómo llegar hasta la torre Eiffel?

Un método podía ser tomar aleatoriamente una calle esperando que más pronto que tarde se llegará a la torre. Esta búsqueda aleatoria puede llevar a encontrar la torre pero puede requerir una cantidad infinita de tiempo por la forma arbitraria en la cual seleccionamos un camino (el mismo puede tomarse múltiple veces).

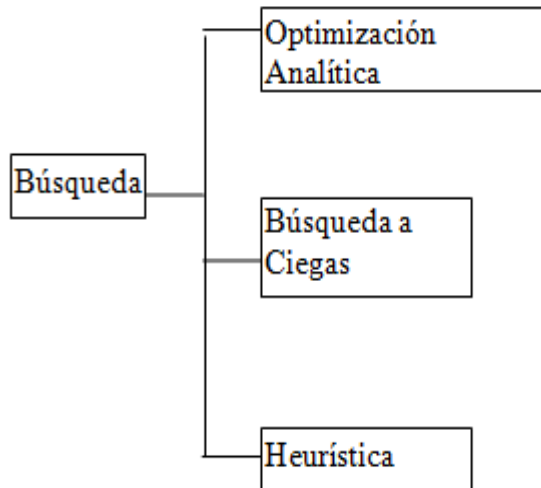
Otra alternativa es seguir exhaustivamente cada calle de inicio a fin. Cuando se alcanza un final se busca una calle paralela y se sigue esta en dirección opuesta independientemente de si nos acercamos o alejamos del objetivo. Eventualmente esta variante consideraría todas las posiciones de nuestro espacio problema. Este tipo de búsqueda se llama a ciegas, ya que no usa conocimiento de cuan cerca estamos de la solución para tomar un determinado camino.

Alternativamente, podemos usar nuestro conocimiento sobre la torre para mejorar la eficiencia de la búsqueda. Suponiendo que el extremo superior de la torre puede ser visto desde cualquier lugar del espacio problema,



podemos tomar la calle que nos parezca nos lleve en esa dirección. Esta es llamada búsqueda dirigida. La búsqueda dirigida es la base de la I.A.

En la siguiente figura se relacionan estas tres alternativas de búsqueda:



**Figura III.8 Alternativas de Búsqueda.**

Otro aspecto a considerar es la dirección de la búsqueda. Puede ser dirigida por dato (forward) o dirigida por objetivo (backward).

Una búsqueda dirigida por dato comienza a partir de la información (o hechos) y trata de extraer conclusiones. Una búsqueda dirigida por objetivo comienza a partir de expectativas de lo que es el objetivo o lo que sucederá, entonces busca las evidencias que soportan o contradicen esas expectativas (o hipótesis).

### **3.3.1 Definición formal de la solución de problemas en la I.A.**

Para resolver un problema primero debemos identificar:

- Un punto de partida.
- Un objetivo a alcanzar.
- Acciones a disposición para resolver el problema.
- Restricciones sobre el objetivo (p.e. de costo).

- Elementos del dominio que son relevantes en el problema (p.e., conocimiento incompleto del punto de partida).

**Estrategias de Búsqueda:** La estrategia de búsqueda define el criterio para seleccionar el siguiente nodo a expandir. La estrategia se evalúa atendiendo a cuatro aspectos:

- **Completitud:** ¿Garantiza la estrategia encontrar una solución cuando esta exista?
- **Complejidad del tiempo:** ¿Cuánto tiempo requiere encontrar una solución?
- **Complejidad del espacio:** ¿Cuánta memoria se necesita para realizar la búsqueda?
- **Optimalidad:** ¿Encuentra la estrategia la solución de mayor calidad cuando haya varias soluciones diferentes?

En general y con mucha frecuencia las estrategias de búsqueda son:

- búsqueda exhaustiva.
- búsqueda a ciegas o no informada.
- búsqueda informada o heurística.

En la búsqueda exhaustiva la idea es examinar el espacio de estado completamente de una manera ordenada, usando todos los operadores y generando todos los sucesores posibles para encontrar la solución deseada. La búsqueda a ciegas es aquella donde no existe ninguna información para decidir que nodo expandir, no se conoce la cantidad de pasos o el costo del camino desde el estado actual hasta el objetivo. También se denomina búsqueda no informada. En el otro caso, cuando existe información para decidir, la búsqueda se denomina informada o heurística.

### **3.3.1.1 Búsqueda a ciega (blind search) o sin información.**

La búsqueda a ciegas es una colección de procedimientos que investigan el espacio de estados de manera exhaustiva pero ciega. Estos procedimientos se consideran métodos débiles, pues imponen restricciones mínimas a la búsqueda, en general son técnicas de solución de problemas de propósito general y pueden describirse independientemente de cualquiera sea el dominio del problema. Estos métodos usan solamente la información estructural y no hacen ninguna distinción cualitativa entre los nodos, respecto a su posibilidad de encontrarse sobre el camino deseado. En consecuencia, para los problemas con un extenso espacio de estados, la cantidad de alternativas que deben explorarse es tan grande que hace que su uso sea computacionalmente imposible. El número de nodos a explorar crece, en general, exponencialmente con la longitud del camino que representa la solución del problema. Esto genera una explosión combinatoria que estos métodos son incapaces de superar. No obstante, continúan formando el núcleo de la mayoría de los sistemas de I.A. [Meseguer, 2006].

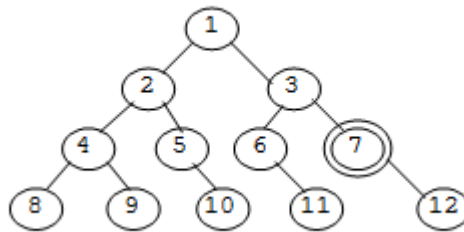
#### **Características:**

- El orden en que la búsqueda se realiza no depende de la naturaleza de la solución buscada.
- Exploración del árbol de búsqueda sistemáticamente pero sin información.
- La localización de la meta(s) no altera el orden de expansión de los nodos.
- No tienen en cuenta el coste de la solución en la búsqueda.
- Su funcionamiento es sistemático, siguen un orden de visitas y generación de nodo establecido por la estructura del espacio de búsqueda.
- Los dos métodos básicos de la búsqueda a ciegas son la búsqueda primero a lo ancho (breadth-first search) y la búsqueda primero en profundidad (depth-first search).

Estudiaremos algunas de las técnicas de búsqueda a ciegas más usadas

### 3.3.1.2 Búsqueda primero a lo ancho (breadth-First)

Una búsqueda primero a lo ancho (breadth-first) explora primero todos los sucesores del nodo raíz. Si no se encuentra la meta, pasa a los sucesores del segundo nivel y así sucesivamente por niveles. Suponiendo que el objetivo a alcanzar es el nodo 7, el recorrido primero a lo ancho del espacio de búsqueda que se muestra en la figura III.9, es 1-2-3-4-5-6-7. Este método simboliza a un explorador bastante conservador.

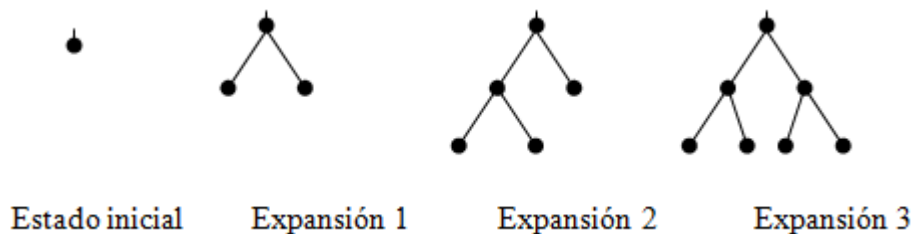


**Figura. III.9. Un ejemplo de espacio de búsqueda.**

Este método tiene como ventaja que siempre encuentra el camino más corto a la solución, si ésta existe, aún en el caso de que el espacio de búsqueda sea infinito. Es efectivo cuando el factor de ramificación, o sea, el número promedio de hijos de un nodo, es pequeño, pues entonces la cantidad de nodos por niveles será pequeña y es mejor explorar un nivel antes de pasar al siguiente. Sin embargo, tiene las siguientes desventajas:

- Necesita mucha memoria. Como cada nivel del árbol tiene que ser almacenado completamente para poder generar el próximo nivel y la cantidad de memoria es proporcional al número de nodos almacenados, su complejidad espacial también lo es.
- Requiere mucho trabajo, especialmente si el camino más corto a la solución es muy largo, puesto que el número de nodos que necesita examinar se incrementa exponencialmente con la longitud del camino.
- Los operadores irrelevantes o redundantes incrementarán grandemente el número de nodos que deben explorarse.

- Una búsqueda primero a lo ancho comienza por generar todos los sucesores de la raíz del nodo. Luego la búsqueda continúa expandiendo todos los nodos de cada nivel. La esencia del método es examinar todas las soluciones potenciales a un nivel antes de pasar al nivel inferior.



**Fig. III.10. Búsqueda Primero a lo ancho.**

### 3.3.1.3 Búsqueda primero en profundidad. (Depth-First).

Este método siempre expande uno de los nodos del nivel más profundo del árbol. Solamente cuando la búsqueda alcanza un nodo muerto (nodo no objetivo que no se puede expandir) la búsqueda va a atrás y expande nodos de niveles inferiores [Morales, 1999].

Esta búsqueda tiene requerimientos de memoria modestos. Solamente almacena un camino de la raíz a un nodo hoja junto con los hermanos de los nodos de este camino a cada nivel. La principal ventaja de este método es que sus requerimientos de memoria son linealmente proporcionales a la profundidad del árbol. La cantidad de tiempo que consume es la misma que en primero a lo ancho.

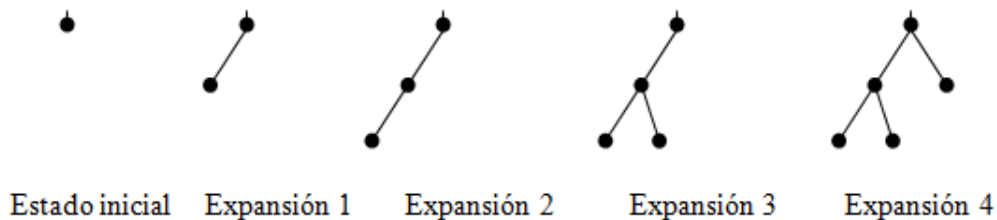
Su principal problema es que puede tomar un camino equivocado que si es muy largo, quizás infinito, no permite encontrar una solución nunca. Muchos problemas tienen árboles de búsquedas muy profundos o incluso infinitos, en cuyo caso este método nunca se podrá recuperar de una selección equivocada en uno de los nodos cercanos a la raíz del árbol.

Otra situación en la que puede caer el método es encontrar una solución con un camino más largo que la solución óptima, es decir, el método no

es ni completo, ni óptimo. Este método debe evitarse para árboles de búsquedas muy profundos. Es un método apropiado cuando los estados objetivos se encuentran a la izquierda del árbol de búsqueda o cuando los caminos son cortos.

En depth-first cada nodo que es explorado genera todos sus sucesores antes de que otro nodo sea explorado. Después de cada expansión el nuevo hijo es de nuevo seleccionado para expansión. Para problemas con muchas soluciones depth-first puede ser más rápido que breadth-first porque existe una buena posibilidad de encontrar una solución después de explorar una pequeña parte del espacio de búsqueda.

La búsqueda primero en profundidad procede generando primeramente un sucesor del nodo raíz y luego un sucesor de este, y continúa extendiendo este camino solamente hasta que termina o se realiza un corte a alguna profundidad, si no se ha alcanzado el objetivo se realiza un retroceso al nivel anterior para generar otro camino.



**Figura III.11 Búsqueda Primero en Profundidad**

#### **Ventajas de la búsqueda en amplitud:**

- No se “pierde” explorando caminos infructuosos que consumen mucho tiempo sin llegar a una solución o de los que no se vuelve nunca (bucles en profundidad).
- Si hay una solución la encuentra. Es más, si hay varias encuentra la óptima.

### **Ventajas de la búsqueda en profundidad:**

- Requiere mucha menos memoria (sólo hay que guardar el camino actual).
- Puede encontrar una solución sin examinar mucho el árbol, sobre todo si hay varios caminos a la solución.

#### **3.3.1.4. Búsqueda heurística o con Información.**

La palabra heurística viene del griego heuriskein que significa descubrir. Ella es también el origen de eureka, que se deriva de la famosa exclamación de Arquímedes heurika (“lo encontré”), que lanzó al descubrir un método para determinar la pureza del oro. Feigenbaum y Feldman la definen así: “Una heurística (regla heurística o método heurístico) es una regla para engañar, simplificar o para cualquier otra clase de ardid, la cual limita drásticamente la búsqueda de soluciones en grandes espacios de solución” [Meseguer, 2006].

La posibilidad de efectuar la búsqueda de una forma más eficiente se basa en el uso de alguna información específica para el problema a solucionar. Esta información, aunque imprecisa e incompleta, permite distinguir cuáles de los nodos dirigen mejor el avance hacia la meta y permite realizar este avance siempre en la dirección que momentáneamente tiene la mejor perspectiva.

Se llama búsqueda heurística a los métodos de búsqueda que usan este tipo de información. Precisamente por esto a estos métodos, generalmente, se les llaman métodos fuertes. Las técnicas heurísticas son como las guías turísticas: son buenas cuando apuntan a direcciones interesantes y son malas cuando apuntan a callejones sin salida. Usando buenas técnicas heurísticas, podemos esperar lograr buenas soluciones a problemas difíciles, como el del vendedor ambulante, en un tiempo menor que el exponencial.

Existen 3 métodos muy conocidos:

- Generación y prueba.
- Ascenso de colina (-> profundidad primero).
- Primero el mejor.

### **3.3.1.5 Generación y Prueba. (GENERATE-AND-TEST)**

1. Generar una posible solución. (Estado o camino).
2. Comprobar para ver si es una solución, mediante comparación con los elementos del conjunto de objetivos aceptables.
3. Si la solución ha sido encontrada salir, de otra manera, retornar al paso.

### **3.3.1.6 Hill Climbing o Escalador de colinas.**

Existe un conjunto de problemas para los cuales el camino que conduce a la solución es irrelevante. Usualmente esto ocurre en aquellos problemas en los que la descripción del estado contiene toda la información necesaria para una solución. La idea general es comenzar con una configuración completa y hacer modificaciones para mejorar su calidad.

El método de búsqueda conocido por ascensión de colinas (hill-climbing) toma su nombre de la semejanza que tiene con un alpinista quien desea alcanzar rápidamente el pico de una montaña, este selecciona la dirección de ascenso mayor a partir de la posición actual.

Con este método la estrategia es repetidamente expandir un nodo, inspeccionar sus sucesores recién generados, y seleccionar y expandir el mejor entre los sucesores sin mantener referencias a los padres.

Este método es una variación de la búsqueda primero en profundidad en la cual se usa la función heurística para estimular la distancia entre el nodo actual y el nodo objetivo [Peraza, 1997].



Los algoritmos de ascenso a colina son típicamente locales, ya que deciden qué hacer, mirando únicamente a las consecuencias inmediatas de sus opciones. Puede que nunca lleguen a encontrar una solución, si son atrapados en estados que no son el objetivo, desde donde no se puede hallar mejores estados, por ejemplo:

1. **Un máximo local**, que es un estado mejor que sus vecinos pero no es mejor que otros que están algo más alejados.
2. **Una meseta**, es un espacio de búsqueda en el que todo un conjunto de estados vecinos tienen igual valor.
3. **Un risco**, que es un tipo especial de máximo local, imposible de atravesar con movimientos simples.

Hay algunas formas que pueden ayudar a solventar estos problemas, aunque no existe garantía:

1. Para evitar máximos locales, regresar a un estado anterior y explorar en una dirección diferente.
2. Para casos de mesetas, dar un salto grande en alguna dirección y tratar de encontrar una nueva sección del espacio de estado.
3. Para los riscos, aplicar dos o más reglas, antes de realizar una prueba del nuevo estado, esto equivale a moverse en varias direcciones a la vez.

Los algoritmos de ascenso a colina, a pesar de explorar sólo un paso adelante, al examinar el nuevo estado pueden incluir una cierta cantidad de información global codificada en la función objetivo o función heurística. [Costa, 2002]

### **3.3.1.7 Algoritmo Best-First (Primero el mejor)**

La búsqueda por el mejor nodo es una forma de combinar las ventajas de las búsquedas en profundidad y a lo ancho en un único método. En cada paso del proceso de búsqueda se selecciona el más prometedor de aquellos nodos que se han generado hasta el momento. Entonces este se expande usando los operadores para generar sus sucesores. Si uno de

ellos es una solución se termina. Si no, todos esos nuevos nodos se añaden al conjunto de nodos generados hasta ese momento.

Se selecciona de nuevo el nodo más prometedor y el proceso continúa. La selección del nodo a expandir es independiente de la posición en que nos encontramos en el árbol de búsqueda y de la posición del nodo más prometedor. Lo que sucede usualmente es que se realiza un poco de búsqueda a profundidad mientras se explora una rama prometedora. En un momento dado esa rama comienza a ser menos prometedora que otras de más alto nivel que se han ignorado hasta ese momento.

El nombre de búsqueda primero el mejor (Best-First Search) es impreciso. Si realmente pudiéramos expandir el mejor nodo primero no sería una búsqueda sino un procedimiento directo al estado objetivo.

Ascenso de colina inspecciona la que parece la mejor trayectoria hasta el final; la búsqueda primero el mejor analiza varias trayectorias a la vez, siempre siguiendo la mejor trayectoria parcial conocida al momento.

Generalmente la búsqueda primero el mejor encuentra trayectorias más cortas a los estados meta.

### **3.3.1.8 Búsqueda con Sistemas Evolutivos.**

Los Algoritmos Genéticos son métodos de búsqueda que recorren el espacio de posibilidades en forma paralela y aleatoria, obedecen a una analogía con la evolución de las especies Darwiniana. En cada etapa, se tiene una población de soluciones posibles para el problema a partir de ésta se genera una nueva población de soluciones mediante operadores que emulan la selección entre las especies del cruzamiento y la mutación. El método trabaja, generación tras generación, mejorando la calidad de la mejor solución de cada población hasta que algún criterio de detección se cumpla, por ejemplo que el esfuerzo computacional que se ha invertido en la solución del problema ha superado el límite predefinido [Tolmos, 2004].

Cuál es el mejor método:

- Primero en profundidad es bueno cuando se sabe con seguridad que el árbol no es muy profundo.
- Primero en anchura, cuando el factor de ramificación no es muy grande.
- Los métodos heurísticos son adecuados cuando existe una medida natural de la distancia entre cada estado y el estado meta.

### **3.4 Resumen Comparativo de las Técnicas I.A.**

Algunos sistemas basados en el conocimiento, como los sistemas Fuzzy encuentran aplicación en automatizar características de percepción, conocimiento y toma de decisiones propias de los operarios humanos. Las redes neuronales artificiales (RNA) emulan ligeramente a las redes neuronales naturales (RNN). Se les ha utilizado, por ejemplo, para aprender la forma de controlar un sistema “observando” las acciones que efectúa un operario. Los algoritmos genéticos (AG) se utilizan en el diseño asistido por computadora para “evolucionar” controladores, bajo el principio de supervivencia para los que mejor se ajusten al objetivo determinado.

### **Conclusiones.**

- En este Capítulo, aprendimos el funcionamiento de las Técnicas de Inteligencia Artificial como son: la lógica difusa la cual es ideal para resolver problemas de la vida real con alto nivel de incertidumbre, debido a sus características de ambigüedad.
- Estudiamos las principales características de las redes neuronales y algoritmos genéticos que son utilizados principalmente para el reconocimiento de voz, imágenes y son métodos de optimización de problemas basados en los mecanismos de la reproducción natural.
- También estudiamos las características básicas de los tipos de búsquedas para resolver problemas de Inteligencia Artificial como: primero el mejor, heurística, primero en amplitud, primero en profundidad.

## **CAPITULO 4. Implementación del Software.**

### **4.1. Introducción.**

El Capítulo muestra la Implementación del software para el Tutorial de Inteligencia Artificial, donde se plantean los requerimientos funcionales que debe asegurar el sistema para satisfacer al cliente y usuarios finales.

Se aborda lo referido al diseño de la Interfaz de Usuario, la Base de Datos, Actores, Casos de Uso, diagramas de Casos de Uso, describiendo cada Caso de Uso textualmente sobre la base de su prototipo de Interfaz-Usuario, realizado con el uso de artefactos del Lenguaje Unificado de Modelado (UML). Se refiere además a la gestión de usuarios del Sistema, definiendo los tipos de Usuario Estudiante y Profesor con sus respectivos permisos de acceder al Sistema y el diagrama de Navegación.

También para la implementación de la Base de Datos conoceremos las características de MySQL, para el diseño de la Interfaz de Usuario se aprenderá las ventajas del entorno de programación PHP, Dreamweaver como editor de páginas Web y como servidor Web el Apache.

### **4.2 Funcionamiento del Sistema**

Este Tutorial brinda Información sobre temas de Inteligencia Artificial, un servicio de Ejercicios Evaluativos, en donde los estudiantes pueden

acceder a la información y evaluarse mientras los profesores pueden consultar y eliminar evaluaciones realizadas por sus estudiantes, también se visualiza el contenido de cada Tema de Inteligencia Artificial. Todas las funcionalidades que tiene la aplicación se basa en información que no debe ser alterada y a la cual no debe tener acceso cualquier tipo de usuario para garantizar que el proceso de gestión sea confiable, por esa razón se decide asegurar la integridad de los datos en la Base de Datos iadatabase, donde se definen tipos y cuentas de usuarios con los cuales se controla el acceso tanto al diseño como modificación de los datos.

#### **4.3 Diseño de la Interfaz-Usuario.**

El diseño de la Interfaz de Usuario debe estar en correspondencia con los requerimientos funcionales que debe cumplir el sistema y satisfacer a los usuarios finales. La opción de elegir las herramientas para la implementación de un sistema es una tarea a veces compleja.

Este Sistema, es una aplicación Web, cuyo objetivo está centrado en lograr una interfaz amigable y legible para el usuario, que sea capaz de cumplir con los requerimientos y dar respuesta en el menor tiempo posible.

Los requerimientos planteados en el Sistema son:

- R1: Consultar Información sobre Inteligencia artificial
- R2: Gestionar Ejercicios Evaluativos.
- R3: Gestionar Cuenta de Usuarios.
- R4: Autenticar al usuario del Sistema.
- R5: Gestionar Documentos Adicionales.
- R6: Evaluar Estudiante.

#### **Ver Anexo IV.1 Interfaz Usuario.**

Para construir este Tutorial elegimos PHP (Personal Home Page por sus siglas en inglés) el cual, es uno de los lenguajes más populares utilizados en la creación de sitios Web dinámicos en Internet además de ser Open

Source e interpretado en el lado del servidor el cual se muestra como código embebido dentro de una página HTML [García,2004].

Se realizó una caracterización más amplia y profunda de PHP por ser este el lenguaje escogido para la implantación del Tutorial de Inteligencia Artificial, resultando que:

- Es un lenguaje específicamente diseñado para realizar aplicaciones Web, mientras que otros lenguajes son adaptaciones de lenguajes preexistentes, no pensados para la Web.
- Mantiene un bajo consumo de recursos de máquina.
- Gran seguridad, muy poca probabilidad de corromper los datos.
- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Informix, entre otras.
- Rico en funciones predefinidas.
- Fácil aprendizaje.
- Es un lenguaje libre.
- Dispone de muchísimos más recursos que ASP.
- Trabaja en combinación con otras tecnologías: perl, javascript, python y dhmtl puedes hacerlo todo, por ejemplo: Creación de gráficos interactivos por el usuario, al estilo photoshop.
- Combina excelentemente con otras inmejorables herramientas, como son el servidor apache y la base de datos mysql (o msql, o postgres), todas ellas gratuitas.
- Buena documentación.
- PHP corre en casi cualquier plataforma utilizando el mismo código fuente, como es Unix, Windows (95,98, NT, ME, 2000, XP) y Macs. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al SO.
- PHP actualmente se puede ejecutar bajo Apache, IIS, AOLServer, Roxen y THTTPD.
- Rapidez.PHP generalmente es usado como modulo de Apache, que lo hace extremadamente veloz. Esta completamente escrito en C.
- Miles de ejemplos y código fuente disponible.

Este Sistema Tutorial de Inteligencia Artificial se publicará en el servidor Web Apache por ser un servidor por excelencia. Apache ha sido uno de los mayores éxitos del software libre. Su continuo desarrollo y portabilidad le han llevado a múltiples plataformas como Windows y casi todos los sistemas Unix.

Principales características de Apache [Pérez, 2004]:

- Es un servidor Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos HTTP.
- Implementa los últimos protocolos aunque se base en HTTP.
- Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches de solución a los mismos.

Dentro de las herramientas para la implementación de la interfaz de usuario del Tutorial de Inteligencia Artificial se encuentra Dreamweaver 8.0, el cual es el más avanzado para el desarrollo de aplicaciones y sitios Web, utiliza un editor de diseño y código de primera calidad en la misma herramienta. A través de sus características y funcionalidades nos permite crear documentos Web que se adapten a nuestras necesidades. En dicho software aunque el usuario que lo maneje sea un experto programador de HTML, siempre encontrarán razones para utilizarlo.

Dreamweaver incluye soporte para la creación de páginas dinámicas de servidor en ASP, PHP, HTML con acceso a bases de datos y una mayor integración con otras herramientas de Macromedia como Fireworks. Este permite editar las páginas Web de forma visual sin tener que teclear el código HTML, además de brindar la opción de trabajar con el código, y por último la posibilidad de ver ambas ventanas de desarrollo a la vez. También posibilita crear botones flash y formularios. Además incluye nuevas herramientas de zoom y guía para revisar los diseños, y una barra de código para acceder funciones frecuentes. Destaca también que las

funciones para cargar y descargar archivos ahora funcionan en el background sin interrumpir la productividad en el programa.

#### **4.4 Lenguaje de Modelado Unificado.**

Para el diseño del Sistema se emplearon algunos de los artefactos del Lenguaje de Modelado Unificado (UML) el cual según la literatura referenciada se ha convertido en el estándar internacional para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación. Su utilización es independiente del lenguaje de programación y de las características del proyecto, pues UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos o de cualquier otra rama. UML se caracteriza por [Booch, 2000]:

- Ser un lenguaje gráfico con una semántica bien definida que estandariza la modelación durante el proceso de desarrollo del software para que sea legible por todo el equipo de proyecto y usuario.
- Construye modelos precisos, no ambiguos y completos.
- No es un lenguaje de programación, pero sus modelos pueden transformarse en código fuente, tablas o almacenamiento de objetos (Generación directa del código).
- Permite describir requerimientos, la arquitectura y modelar las pruebas a través de artefactos que permiten documentar el proceso.

Es importante recalcar que sólo se trata de una notación, es decir, de una serie de reglas y recomendaciones para representar modelos. UML no es un proceso de desarrollo, no describe los pasos sistemáticos a seguir para desarrollar el software. UML sólo permite documentar y especificar los elementos creados mediante un lenguaje común describiendo modelos.

Por las ventajas antes referidas para el diseño de la Interfaz de Usuario del Tutorial de Inteligencia Artificial se emplearon algunos de los artefactos del Lenguaje de Modelado Unificado (UML),



#### 4.5. Casos de Uso.

En el diseño del tutorial se utilizaron específicamente vistas lógicas para representar las funcionalidades a realizar por el Sistema, empleándose los conceptos de:

- **Actores:** elementos que interactúan con la aplicación ya sea un humano, un software o hardware.
- **Casos de usos:** agrupación de fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para los actores.
- **Diagrama de Caso de Uso:** modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado. Se representa la relación entre los casos de uso y los actores relacionados con estos.

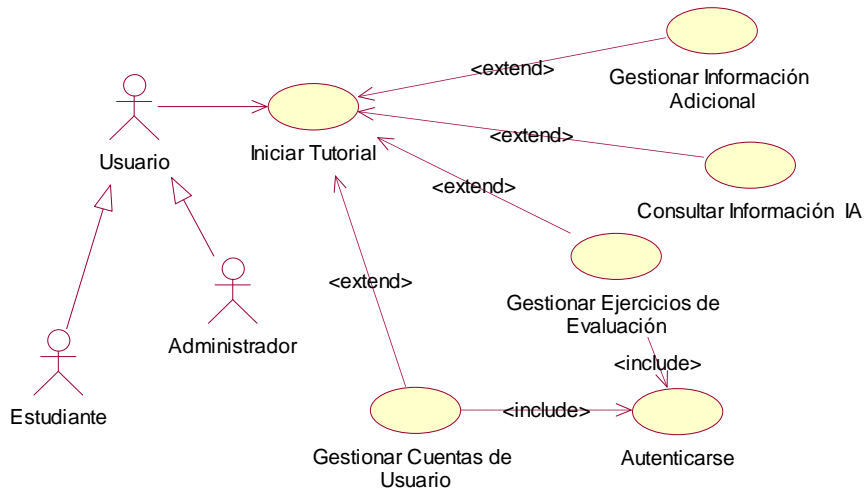
En el Tutorial de Inteligencia Artificial se definieron los tipos de usuario siguientes:

- **ADMINISTRADOR/PROFESOR** tiene el derecho de controlar toda la información brindada por el sistema incluyendo la del registro de usuarios, puede controlar la Información Adicional sobre Inteligencia Artificial y los Ejercicios Evaluativos.
- **ESTUDIANTE** solo tiene el derecho de poder leer la información de la aplicación y realizar los ejercicios evaluativos propuestos.
- **USUARIO** puede consultar información como es: Información sobre Inteligencia Artificial y todo su contenido.

<b>Actor</b>	<b>Rol</b>
Usuario/Invitado	Consultar la información sobre Inteligencia Artificial, así como los Capítulos Teóricos
Estudiante	Realizar los Ejercicios Evaluativos propuestos.
Profesor/ Administración	Controlar la Información sobre Inteligencia Artificial, Los Ejercicios Evaluativos y Registro de Usuarios.

**Tabla IV.1. Actores del Tutorial de Inteligencia Artificial.**

Los Casos de Uso definidos para el sistema se encuentran relacionados dentro del Diagrama de Caso de Uso con el objetivo de lograr una mayor y mejor comprensión del Modelo y modularización de la funcionalidad que brinda el sistema estos se muestran en la figura IV.1, cada uno de los casos de uso están relacionados entre sí representando el flujo de información o comunicación que existe entre ellos.



**Figura IV.1- Diagrama de Casos de Uso del Sistema.**

Se han definido siete Casos de Uso que garantizan un análisis modular del sistema. Los Casos de Uso son: Iniciar Tutorial, Gestionar Cuenta de Usuario, Consultar Información sobre Inteligencia Artificial, Gestionar Ejercicios Evaluativos, Autenticarse, Gestionar Información Adicional y Evaluarse. Tabla IV.2.

Casos de Uso	Requerimientos Asociados	Clasificación
Iniciar Tutorial		Principal
Gestionar Cuentas de Usuario	R3,R4	Principal
Consultar Información	R1	Principal
Gestionar Ejercicios Evaluativos	R2,R4	Principal
Autenticarse	R4	Principal
Gestionar Información Adicional	R5,R4	Principal
Evaluar Estudiante	R6	Principal

**Tabla IV.2 Casos de Uso del Sistema**

A continuación se describe textualmente cada uno de los Casos de Uso del Diagrama empleado su prototipo de su Interfaz-Usuario.

<b>Caso de uso:</b>	<b>Iniciar Tutorial.</b>
<b>Actores:</b>	Usuario(inicia)
<b>Descripción:</b>	Este caso de uso se inicia cuando un usuario desea acceder a la información que contiene el Tutorial y para eso necesita iniciar la aplicación.
<b>Referencias:</b>	
<b>Precondiciones:</b>	
<b>Poscondiciones:</b>	El acceso al Tutorial
<b>Curso normal de los eventos</b>	
<b>Acción del Usuario</b>	<b>Respuesta del Sistema</b>
1. Accede al Tutorial	2. Muestra la interfaz principal de la aplicación (IAP).

Tabla IV.3.Caso de Uso Iniciar Tutorial.



Figura IV.2 Figura de Iniciar Tutorial (IP).

<b>Caso de uso:</b>	<b>Autenticarse</b>
<b>Actores:</b>	Estudiante, Profesor/ Administrador (inician).
<b>Descripción:</b> El Caso de Uso se inicia cuando el usuario necesita hacer uso del sistema Tutorial, una vez realizada su autenticación concluye el Caso de Uso. En caso de ser positiva la autenticación se notifica la bienvenida al sistema y en caso contrario no se permite la entrada al Sistema.	
<b>Referencias:</b>	R4
<b>Precondiciones:</b>	El usuario haya iniciado el Sistema
<b>Poscondiciones:</b>	Que se ha iniciado o no la sesión del usuario
<b>Curso normal de los eventos</b>	
<b>Acción del usuario</b>	<b>Respuesta del Sistema</b>
3. Introduce su Usuario en (A1 ), la contraseña en (A2) y presiona el botón Aceptar (A3)	4. Muestra el tipo de usuario en pantalla, las acciones que puede realizar este tipo de usuario y se notifica la entrada al sistema.

**Tabla IV.4. Caso de Uso Autenticarse**

The image shows a screenshot of a web interface titled 'Autenticación'. It features three input elements: a text box labeled 'Usuario' with the identifier 'A1' to its right, a text box labeled 'Contraseña' with the identifier 'A2' to its right, and a button labeled 'Aceptar' with the identifier 'A3' to its right. The interface has a blue header bar with the title 'Autenticación'.

**Figura IV. 3- Figura Interfaz Autenticarse.**

<b>Caso de uso:</b>	<b>Consultar Información sobre temas I.A.</b>
<b>Actores:</b>	Usuario (inicia).
<b>Descripción:</b>	El Caso de Uso se inicia cuando el usuario consulta: Información sobre temas de Inteligencia Artificial
<b>Referencias:</b>	R1
<b>Precondiciones:</b>	El usuario haya iniciado el tutorial.
<b>Poscondiciones:</b>	La Información sobre Inteligencia Artificial ha sido consultada.
<b>Curso normal de los eventos</b>	
<b>Acción del usuario</b>	<b>Respuesta del Sistema</b>
1. En el menú principal de la Interfaz del tutorial selecciona el tema sobre Inteligencia Artificial.	2. Muestra en pantalla la interfaz consultar Información sobre Inteligencia Artificial.
3. Selecciona en la interfaz (CIIA1) la información que desee consultar.	Muestra la Información consultada en (CIIA2).

**Tabla IV.5. Caso de Uso Consultar Información sobre I.A.**

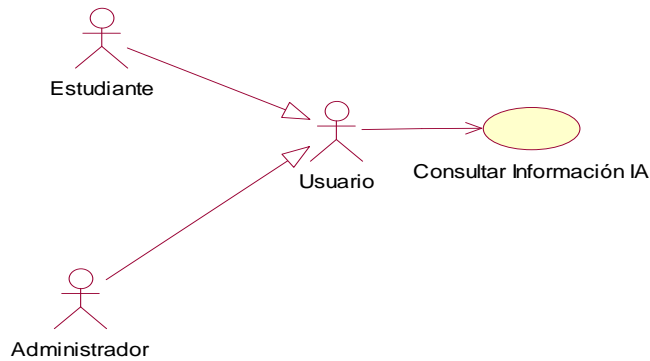
**Sitio WEB para la asignatura de Inteligencia Artificial**

14 de Enero de 2009

<b>Tema I "Surgimiento y Desarrollo"</b>	<b>Conceptos</b>	<b>Autenticación</b>
<ul style="list-style-type: none"> <li>Inteligencia</li> <li>Conceptos</li> <li>Relacion con otras ciencias</li> <li>Paradigmas de IA</li> <li>Tendencias de la IA</li> </ul>	<p><b>CIIA2</b></p> <p>Aprendizaje e Inteligencias</p> <p>Dado que el Aprendizaje es el aspecto principal de cualquier concepción teórica de la Inteligencia, no resulta sorprendente que se haya prestado mucha atención a las relaciones entre estos dos conceptos. Tres preguntas son básicas para comprender la relación entre Aprendizaje e Inteligencia.</p> <p>Ellas son las siguientes:</p> <ul style="list-style-type: none"> <li>¿Cómo surge la Inteligencia?</li> <li>¿Cómo influye la Inteligencia sobre el Aprendizaje?</li> <li>¿Cómo está relacionada la Inteligencia con la eficiencia del Procesamiento de la Información que sigue al Aprendizaje?</li> </ul> <p>De alguna forma ya hemos respondido a las interrogantes anteriores. Hemos visto que a la inteligencia se le ha interpretado en dependencia del aprendizaje y de la experiencia. Hemos visto también que hay un reconocimiento de algunos rasgos del comportamiento que la manifiestan, y uno de ellos es precisamente la capacidad de</p>	<p>Usuario</p> <input type="text"/> <p>Contraseña</p> <input type="text"/> <input type="button" value="Aceptar"/>
<b>Tema II "Programacion Logica"</b>		
<ul style="list-style-type: none"> <li>Programacion descriptiva</li> <li>Prolog</li> <li>Características de Prolog</li> <li>Tipos de datos</li> <li>Listas en Prolog</li> <li>El predicado padre</li> <li>El predicado miembro</li> </ul>		

**CIIA1**

**Figura IV.4 - Interfaz Consultar Información sobre I.A. (CIIA).**



**Figura IV.5- Caso de Uso Consultar Información sobre I.A.**

<b>Caso de uso:</b>	<b>Gestionar Información Adicional I.A.</b>
<b>Actores:</b>	Profesor/Administrador (inician).
<b>Descripción:</b> El Caso de Uso puede ser iniciado por el profesor o por el administrador una vez que decidan administrar la Información Adicional sobre Inteligencia Artificial Para realizar esta acción tienen que antes haberse autenticado.	
<b>Referencias:</b>	R5
<b>Precondiciones:</b>	Que el usuario este autenticado como administrador o como profesor.
<b>Poscondiciones:</b>	Los temas de Información Adicional sobre Inteligencia Artificial han sido Administrados.
<b>Curso normal de los eventos</b>	
<b>Acción del usuario</b>	<b>Respuesta del Tutorial</b>
1. Necesita agregar un archivo al Tutorial IA	2. Muestra la Interfaz Autenticación (IAPC)
3. El usuario introduce: usuario en (A1) y contraseña en (A2).	4. Muestra listado de temas de Información Adicional sobre Inteligencia Artificial (IAIA) que existen.
5. Selecciona el nombre (NAI) y ruta de archivo (IAIA3) a anexar o eliminar (EDA) y ejecuta (IAIA2)	6. Muestra la ruta de ubicación de la Información para Administrarla en (IAIA3).
7. Acepta y Actualiza la Información en (IAIA) y se coloca en el menú Principal para acceder estudiante	8. Visualiza la Información Actualizada en (IAIA4). Para que usuario acceda a la información y guarda en la base de datos.

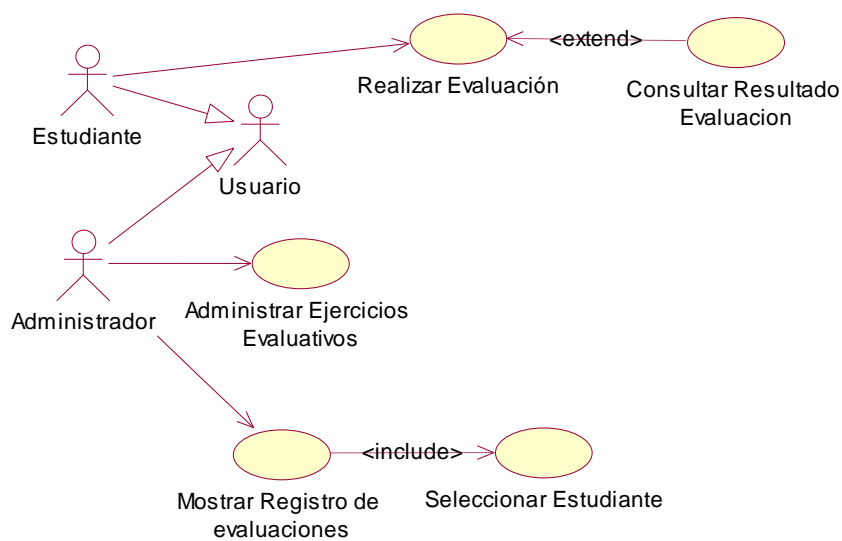
**Tabla IV.6. Caso de Uso Gestionar Información Adicional.**





**Figura IV. 6. Gestión de Documentos Adicionales**

La descripción del Caso de Uso **Gestionar Ejercicios Evaluativos**, se resume en la descripción de los casos de uso: Realizar Evaluación, Consultar Resultados de Evaluaciones, Mostrar Registro de Evaluaciones, Seleccionar Estudiante, Eliminar Evaluación, Administrar ejercicios evaluativos y Autenticarse. En la figura IV.7 se muestra el Caso Uso Gestionar Ejercicios Evaluativos.



**Figura IV.7 - Caso Uso Gestionar Ejercicios Evaluativos.**

<b>Caso de uso:</b>	<b>Realizar Evaluación.</b>
<b>Actores:</b>	Estudiante
<b>Descripción:</b> El Caso de Uso es iniciado cuando los usuarios del sistema desean realizar una evaluación a través de los Ejercicios Evaluativos.	
<b>Referencias:</b>	R2, R4
<b>Precondiciones:</b>	Que el usuario este autenticado como Usuario y la Aplicación este iniciada.
<b>Poscondiciones:</b>	Que se realice la evaluación.
<b>Curso normal de los eventos</b>	
<b>Acción del usuario</b>	<b>Respuesta del Sistema</b>
1. En el menú principal de la Interfaz Principal se Autentifica como usuario estudiante y selecciona la opción Auto evaluación (RE1).	2. Muestra la interfaz Ejercicios Evaluativos(EV)
3. Selecciona el tema de evaluación en (RE2).	4. Muestra el texto de pregunta del tema seleccionado en (RE3) y las posibles respuestas de dicha pregunta en (RE4).
5. Marca las respuesta que considera correctas en (RE4) y realiza la evaluación presionando el botón Responder (RE5),en caso que no desee evaluarse en una pregunta determinada puede pasar para la próxima o la anterior en (RE6)	6. Guarda los resultados en la Base de Datos del Tutorial y Muestra en pantalla la interfaz Resultado de Evaluaciones y los resultados de las evaluaciones realizadas por el usuario.

**Tabla IV.7. Caso de Uso Realizar Evaluación.**



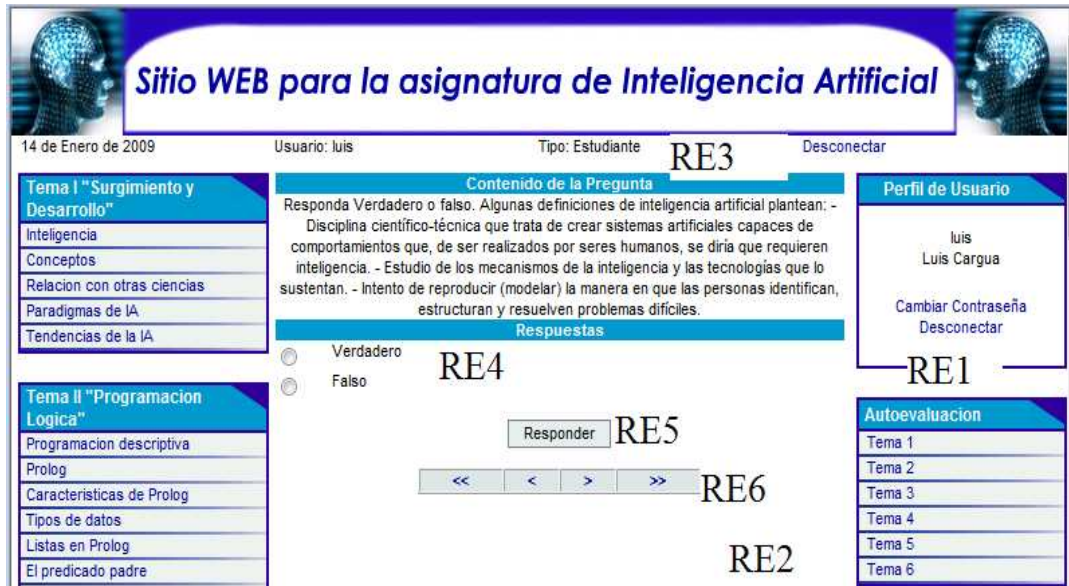
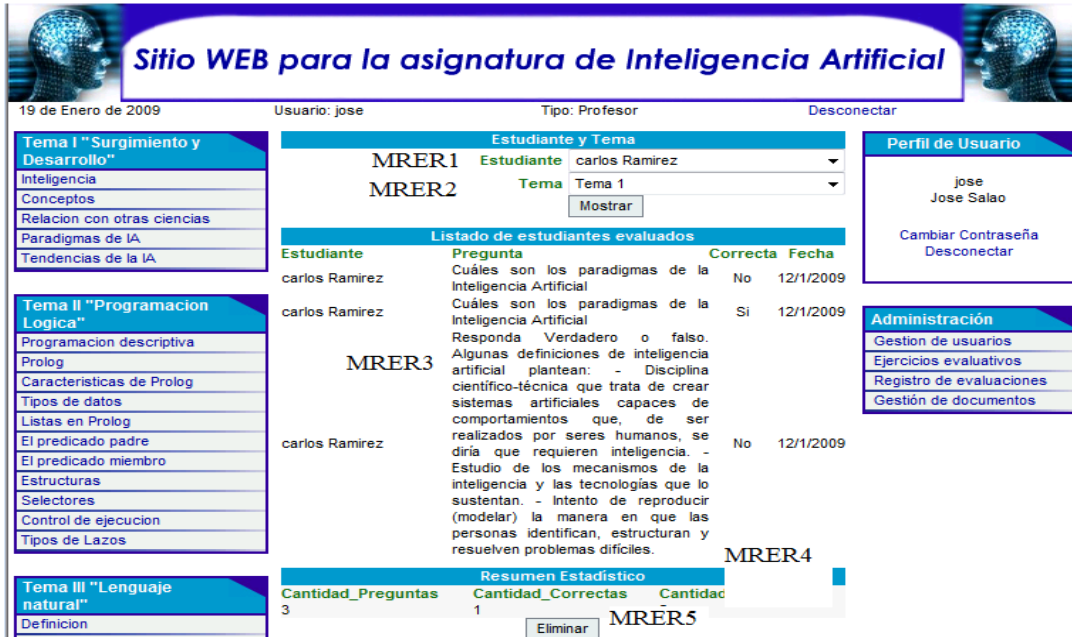


Figura IV.8- Interfaz Realizar Evaluación (RE)

<b>Caso de uso:</b>	<b>Mostrar Registro de Evaluaciones Realizadas.</b>
<b>Actores:</b>	<b>Administrador/Profesor (inicia).</b>
<b>Descripción:</b>	El Caso de Uso se inicia cuando el Administrador o Profesor desean obtener los resultados de las evaluaciones realizadas por los Estudiantes.
<b>Referencias:</b>	R2, R4
<b>Precondiciones:</b>	Que el usuario este autenticado como administrador o como profesor.
<b>Poscondiciones:</b>	Que se muestren los resultados de evaluaciones de los Estudiantes.
<b>Curso normal de los eventos</b>	
<b>Acción del usuario</b>	<b>Respuesta del Sistema</b>
1. Ejecuta en la interfaz Registro de Evaluación.	2. Muestra en pantalla la interfaz Registro de Evaluaciones Realizadas (MRER).
3. Selecciona en MRER1 el Estudiante y en MRER2 el tema si lo desea.	4. Se muestra en MRER3 todos los datos de las evaluaciones del estudiante. En (MRER4) se muestra un resumen estadístico con el total de preguntas respondidas y la cantidad correctas e incorrectas y también se activa el botón (MRER5) para si desea eliminar el registro de ese estudiante.
5. Si va a eliminar el estudiante presiona el botón (MRER5)	6. Se elimina el registro de evaluación del estudiante

Tabla IV.8. Caso de Uso Registro Evaluaciones Realizadas.

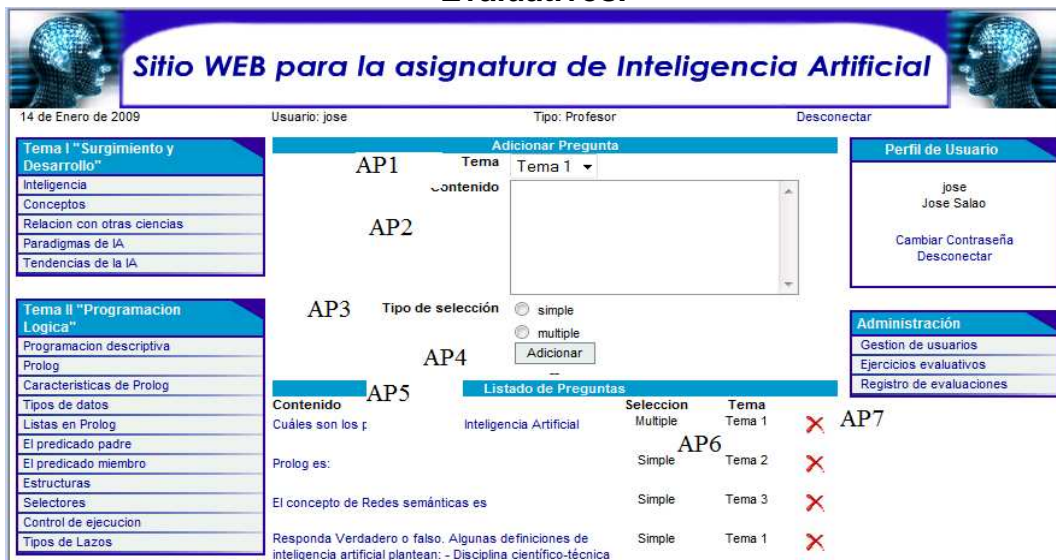


**Figura IV.9- Interfaz Mostrar Registro de Evaluaciones Realizadas (MRER)**

<b>Caso de uso:</b>	<b>Administrar Ejercicios Evaluativos.</b>
<b>Actores:</b>	Administrador/Profesor (inicia).
<b>Descripción:</b>	El Caso de Uso se inicia cuando el Administrador desea administrar los Ejercicios Evaluativos, donde se incluye la administración de los Temas, Preguntas y Respuestas.
<b>Referencias:</b>	R2,R4
<b>Precondiciones:</b>	Que el usuario este autenticado como administrador o como profesor.
<b>Poscondiciones:</b>	Que los Ejercicios Evaluativos hayan sido administrados.
<b>Curso normal de los eventos</b>	
<b>Acción del usuario</b>	<b>Respuesta del Sistema</b>
1. En la interfaz Ejercicios Evaluativos se realiza la Administración de todas las Preguntas relacionadas con el Tema.	2. Muestra la Interfaz Administrar Preguntas (AP).
Puede: En la Interfaz (AP)	Responde en caso de:
<ul style="list-style-type: none"> <li>. <b>Adicionar</b> selecciona en (AP1) el tema, introduce el texto de la Pregunta en (AP2), en (AP3).el tipo de pregunta y con el uso del botón(AP4)</li> <li>. <b>Editar</b> presiona</li> <li>. <b>Modificar</b> con el uso del</li> </ul>	<ul style="list-style-type: none"> <li>. Adicionar, adiciona la pregunta y la muestra en(AP5)</li> <li>. Editarla, muestra los valores en AP1, AP2, AP3 y el botón (AP4) se activa para Modificar.</li> <li>. Modificar, modifica los datos de la pregunta y</li> </ul>

<p>botón (AP6) que se activara según la acción</p> <ul style="list-style-type: none"> <li>. <b>Eliminar</b> presionar AP7</li> <li>. <b>Vínculo a Respuesta</b> ejecuta en AP5 donde está el nombre de la pregunta.</li> </ul>	<p>los muestra modificados en AP5.</p> <ul style="list-style-type: none"> <li>. Eliminar, elimina la Pregunta.</li> <li>. Vínculo a Respuesta, muestra la interfaz de Administrar Respuesta (AR).</li> <li>. Conexión para Administrar Tema, muestra la Interfaz Administrar Temas (AP1).</li> </ul>
<p>Puede: En la interfaz (AR)</p> <ul style="list-style-type: none"> <li>. <b>Adicionar</b> introduce el código de la Respuesta en (AR1), en (AR2) el texto, en (AR3) si es correcta o no y presiona el botón (AR4) para adicionarla.</li> <li>. <b>Editar</b> presiona la imagen (AR6)</li> <li>. <b>Modificar</b> con el uso del botón (AR4) que se activara según la acción.</li> <li>. <b>Eliminar</b> presionar la imagen (AR7).</li> </ul>	<p>Responde en caso de:</p> <ul style="list-style-type: none"> <li>. Adicionar, adiciona la respuesta y la muestra en (AR5).</li> <li>. Editar, muestra los valores en AR1, AR2, AR3 y el botón (AR4) se activa para Modificar.</li> <li>. Modificar, modifica los datos de la Respuesta y los muestra (AR5).</li> <li>. Eliminar, elimina la Respuesta.</li> </ul>

**Tabla IV.9. Caso de Uso Administrar Ejercicios Evaluativos.**



**Figura IV. 10- Interfaz Administrar Pregunta (AP).**



Figura IV. 11- Interfaz Administrar Respuesta (AR).

Descripción del caso de uso **Gestionar Cuenta de Usuario.**

<b>Caso de uso:</b>	<b>Gestionar Cuenta de Usuarios.</b>
<b>Actores:</b>	Administrador (inicia).
<b>Descripción:</b>	El Caso de Uso se inicia cuando el administrador desea administrar las Cuentas de los Usuarios del Tutorial.
<b>Referencias:</b>	R3
<b>Precondiciones:</b>	El usuario este autenticado como administrador.
<b>Poscondiciones:</b>	
<b>Curso normal de los eventos</b>	
<b>Acción del usuario</b>	<b>Respuesta del Sistema</b>
1. En el menú principal de Interfaz del Sistema selecciona la opción Gestión de usuarios	2. Muestra en pantalla la interfaz de Autenticación.
3. El usuario se autentifica como administrador	4. Muestra la interfaz de Gestionar Cuentas de Usuarios (GCU).
. <b>Adicionar</b> introduce en (GCU1) el Nombre Completo del Usuario, en (GCU2) el Usuario, en (GCU3) la Contraseña, en (GCU4) selecciona el Tipo de Usuario	Responde en caso de: . Adicionar, adiciona un Usuario y en (GCU6) muestra sus datos. . Editar, muestra los valores en GCU1, GCU2, GCU3 , GCU4 y en el botón GCU5 se activa

<p>y con el uso del botón (GCU5) adiciona.</p> <ul style="list-style-type: none"> <li>. <b>Editar</b> presiona.</li> <li>. <b>Modificar</b> con el uso del botón (GCU5) que se activara según la acción.</li> <li>. <b>Eliminar</b> presionar la imagen (GCU8).</li> </ul>	<p>para Modificar</p> <ul style="list-style-type: none"> <li>. Modificar, modifica los datos Usuario y los muestra modificados en (GCU6).</li> <li>. Eliminar, elimina el Usuario.</li> </ul>
--	---

**Tabla IV.10. Caso de Uso Gestionar Cuentas de Usuario.**

14 de Enero de 2009      Usuario: jose      Tipo: Profesor      Desconectar

**Tema I "Surgimiento y Desarrollo"**

- Inteligencia
- Conceptos
- Relacion con otras ciencias
- Paradigmas de IA
- Tendencias de la IA

**Adicionar Nuevo Usuario**

Nombre Completo  GCU1

Usuario  GCU2

Contraseña  GCU3

Tipo de Usuario  GCU4

GCU5

**Perfil de Usuario**

jose  
Jose Salao

**Tema II "Programacion Logica"**

- Programacion descriptiva
- Prolog
- Caracteristicas de Prolog
- Tipos de datos
- Listas en Prolog
- El predicado padre
- El predicado miembro
- Estructuras
- Selectores
- Control de ejecucion

**Listado de Usuarios**

Nombre Completo	Usuario	Tipo	GCU7	GCU8
carlos Ramirez	carlos	Estudiante		
Jose Salao	jose	Profesor		
Marcos	marco	Estudiante		
Pepe Antonio	pepe	Estudiante		
Luis Cargua	luis	Estudiante		
Eduardo Heredia	eduardo	Estudiante		
Mario Mora	mario	Estudiante		
Angel Martinez	angel	Estudiante		

**Administración**

- Gestion de usuarios
- Ejercicios evaluativos
- Registro de evaluaciones

**Figura IV.12- Interfaz Gestionar Cuenta Usuario (GCU).**

#### **4.6. Diseño de la Base de Datos del Sistema.**

Para facilitar las tareas de administración de los datos y acelerar el desarrollo de la aplicación se hace necesario realizar una selección adecuada del Sistema Gestor de Bases de Datos (SGBD) que no es más que un conjunto de datos relacionados entre sí y un grupo de programas para tener acceso a esos datos. A continuación se presenta una valoración, en este caso, de MySQL versión 5.0.45 utilizada para implementar la Base de Datos de este Tutorial de I.A.

Teniendo en cuenta la literatura referenciada podemos destacar que MySQL es el servidor de base de datos "Open Source" más utilizado en todo el mundo. Sus principales características son la velocidad, la robustez, aprovechamiento de potencia de sistemas multiprocesador gracias a su implementación multihilo, soporta gran cantidad de tipos de datos, dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc.), gestión de usuarios y contraseña manteniendo un muy buen nivel de seguridad en los datos y además de ser multiplataforma. Es un sistema que se integra a una gran cantidad de sistemas operativos como Windows, AIX, HP-UX systems, Linux Intel, Sun Solaris entre otros.

Para hacer el manejo de MySQL un poco más amigable, se utilizó la herramienta PHPMyAdmin, que es un terminal vía Web para el acceso a las bases de datos y sus tablas. Una vez instalado el programa basta con poner en el navegador <http://localhost/phpmyadmin> y se nos pedirá un usuario root y una contraseña para acceder a las base de datos. Permitido el acceso se tiene la posibilidad de crear las tablas, eliminarlas, consultarla, insertar las filas, modificarlas y varias opciones más muy cómodas. En la figura IV.13 se muestra la interfaz de programa PHPMyAdmin.





Figura IV. 13- Interfaz PHP MyAdmin.

La base de datos de nuestro Sistema permite almacenar la información de forma organizada y detallada, al mismo tiempo permite interactuar entre los diferentes procesos que se pueden generar como consultas y reportes que requiere el usuario. En la figura IV.14 aparecen las entidades que conforman el dominio del problema, sus atributos y relaciones existentes entre ellas.

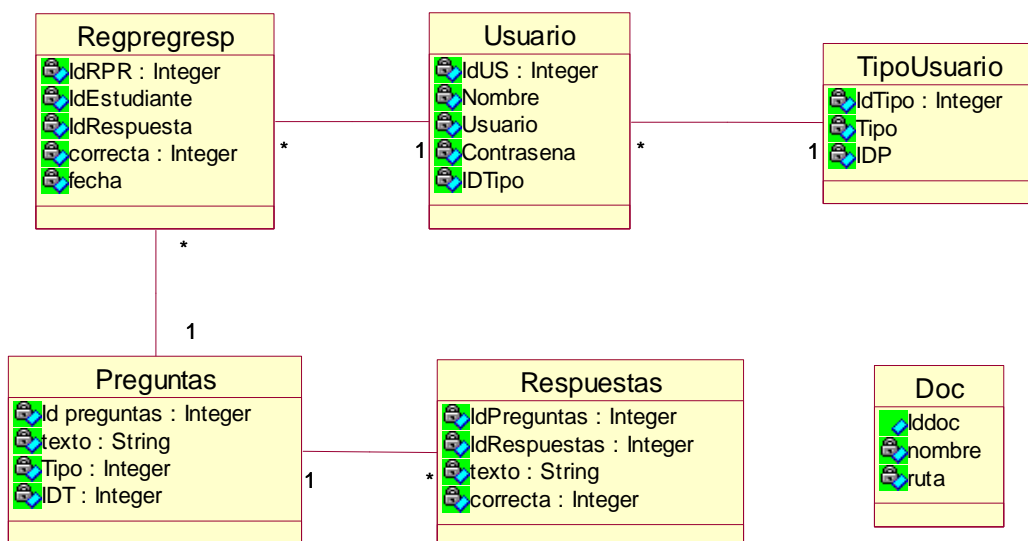


Figura IV.14- Modelo de datos del Tutorial de Inteligencia Artificial

## **Seguridad**

Para asegurar la integridad de los datos en la Base de Datos del Tutorial de IA se utilizó la opción de seguridad a nivel de usuario, pues constituye el modo más fuerte y flexible de protección de una aplicación. Dicha seguridad es implementada a través de una tabla Usuario donde la contraseña del usuario es encriptada por medio de un algoritmo matemático.

Esta aplicación tiene una capa de seguridad en la que se verifica al usuario si está registrado o no. Cuando el usuario ha sido ingresado puede tener privilegios como administrador o como estudiante usuario del Sistema, en caso contrario solamente ingresa al sitio como invitado para consulta de información del Tutorial de Inteligencia Artificial.

**Usuario Administrador.**-Persona que puede realizar la administración de la aplicación.

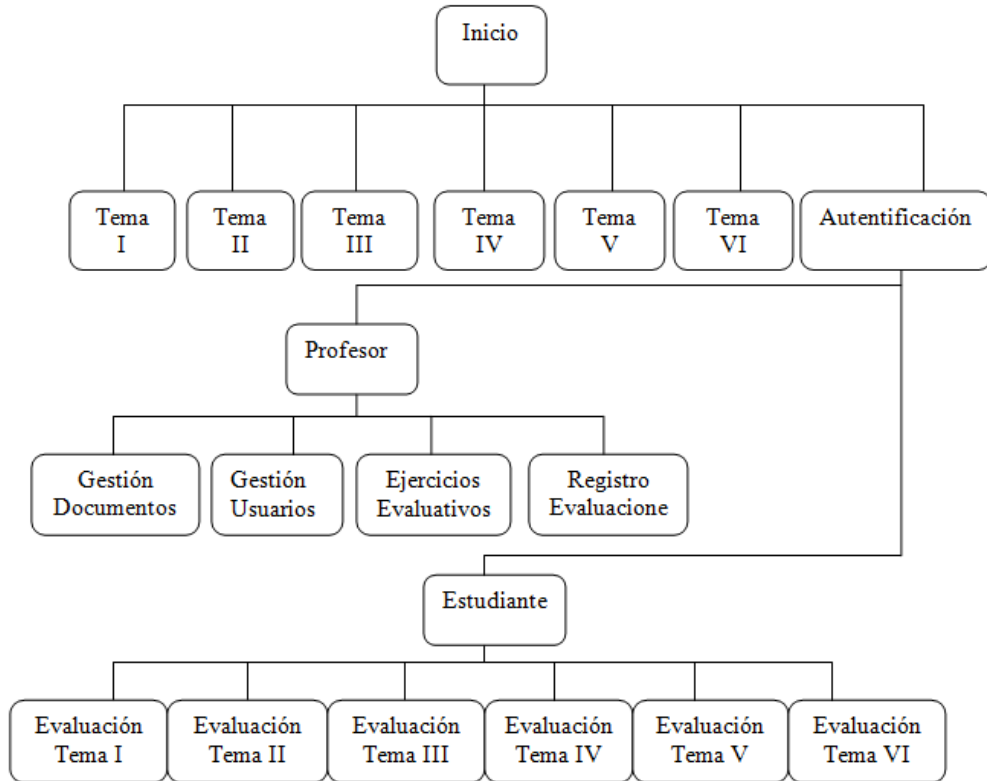
**Usuario Estudiante.**-Persona que puede visualizar la información y puede evaluarse en los temas de la materia.

**Usuario Invitado.**-Persona que puede solamente visualizar la información del tutorial.



#### 4.7. Mapa de Navegación

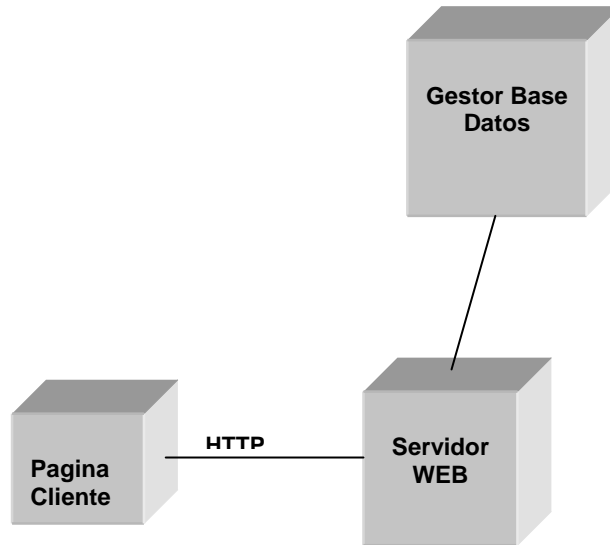
A continuación se muestra el mapa de navegación, a través del cual se visualiza una panorámica de la navegación del sitio.



**Figura IV.14- Mapa de Navegación del Tutorial IA.**

#### 4.8 Modelo de Despliegue.

El Modelo de Despliegue define la arquitectura física del sistema por medio de nodos interconectados. Se utiliza para comprender las actividades de diseño e implementación debido a que la distribución del sistema permite un mejor desarrollo del diseño.



**Figura IV-16- Modelo de Despliegue del Tutorial IA.**

La aplicación que se propone está basada sobre una arquitectura cliente - servidor representada por tres nodos. El nodo "Página Cliente" está representado por un nodo ordenador el cual contiene un navegador para Internet. Este recibe la información en lenguaje HTML enviado desde el servidor y se encarga de comunicarse con el nodo que contiene la aplicación Web a través del protocolo HTTP. Este proceso se realiza a través de los recursos que se le muestran al usuario en la página, esto permite al usuario establecer un sistema de comunicación con el servidor Web Apache.

En el nodo Servidor Web Apache se atienden las solicitudes del cliente, se analizan y se les da respuesta. En este nodo están contenidos todos los procesos de información para garantizar el funcionamiento del servidor logrando cumplir con todos los requerimientos funcionales del sistema. La capa de acceso a datos se comunica con el nodo Gestor de Bases de Datos "MySQL" a través del protocolo TCP/IP donde se encuentra la información almacenada en la base de datos.

#### 4.9 Análisis de Factibilidad

Para la estimación del costo se calcularon los indicadores siguientes con uso del software USC Cocomo II del Centro para Ingeniería del software de la Universidad de California.

**Entradas Externas (EI):** entrada de usuario que proporciona al software diferentes datos orientados a la aplicación.

Nombre	Cantidad de ficheros	Cantidad de Elemento de datos	Complejidad
Adicionar Usuarios	2	6	Bajo
Modificar Usuarios	2	6	Bajo
Eliminar Usuarios	2	6	Bajo
Adicionar Ejercicio Evaluativo	3	3	Medio
Eliminar Ejercicio Evaluativo	3	5	Medio
Adicionar Respuesta	2	8	Medio
Modificar Respuesta	2	8	Medio
Eliminar Respuesta	2	8	Medio
Adicionar Documento Complementario	2	2	Bajo
Eliminar Documento Complementario	1	1	Bajo

**Tabla IV.17- Entradas Externas**

**Salidas Externas (EO):** salida que proporciona al usuario información orientada de la aplicación. En este contexto la “salida” se refiere a informes, pantallas, mensajes de error, etc.

Nombre	Cantidad de ficheros	Cantidad de Elemento de datos	Complejidad
Tema I "Surgimiento y Desarrollo"	1	7	Medio
Tema II "Programación Lógica"	1	3	Bajo
Tema III "Lenguaje natural"	1	3	Bajo
Tema IV "Procesos de búsqueda "	1	5	Bajo
Tema V "Representación del conocimiento "	1	3	Bajo
Tema VI "Redes neuronales"	1	2	Bajo
Documentos Adicionales	2	2	Bajo
Ejercicios Evaluativos	3	3	Bajo
Resultados de Evaluación	3	4	Bajo
Registro de Evaluaciones	3	7	Medio

**Tabla IV.18- Salidas Externas**

**Ficheros internos (ILF):** son archivos (tablas) maestros lógicos (o sea una agrupación lógica de datos que puede ser una parte de una gran base de datos o un archivo independiente).

Nombre	Cantidad de registro	Cantidad de Elemento de datos	Complejidad
Usuario	30+	5	Medio
TipoUsuario	2+	3	Medio
Preguntas	50+	4	Medio
Respuestas	150+	4	Medio
RegPregResp	250+	5	Medio
Doc	15+	3	Medio

**Tabla IV.19- Ficheros Internos**

Se consideró como entorno de programación PHP tomándose como promedio 20 líneas código en este lenguaje por punto de función (según tabla de reconciliación de métricas consultada), obteniéndose así 4940 instrucciones fuentes con un Total de Puntos de Función Desajustados de 247.

**Esfuerzo (DM).**

$$DM = (\text{Valor Optimista} + 4 \times (\text{Valor Esperado}) + \text{Valor Pesimista}) / 6$$

$$DM = (7.6 + 4 * 11.4 + 17.6) / 6 = \mathbf{11.8 \text{ Hombres / mes.}}$$

**Tiempo (TDev).**

$$TDev = (\text{Valor Optimista} + 4 \times (\text{Valor Esperado}) + \text{Valor Pesimista}) / 6$$

$$TDev = (3.7 + 4 * 4.8 + 10.1) / 6 = \mathbf{5.51 \text{ Meses.}}$$

**Cantidad de hombres (CH):**

$$CH = DM / TDev$$

$$CH = 11.8 / 5.51$$

$$CH = 2.14 \text{ hombres}$$

**Costo de la Fuerza de Trabajo.**

$$CTP = (\text{Valor Optimista} + 4 \times (\text{Valor Esperado}) + \text{Valor Pesimista}) / 6$$

$$CTP = (669.26 + 4 * 1337.70 + 1956.56) / 6 = \mathbf{\$1328.97}$$

**4.10 Implementación del Software**

**Recursos Técnicos:**

❖ **Hardware para su diseño y desarrollo:**

Procesador: Pentium IV o superior.

Memoria: 256 Mbytes o superior

Disco Duro: 120 Gbytes

Monitor: Resolución SVGA (800 x 600) píxeles.

❖ **Software:**

Sistema Operativo Windows 2000 o Superior.

MySql Server.

PHPMYAdmin

Lenguaje de script para web PHP.

Servidor de paginas Web Apache.

Macromedia Dreamweaver 8.0

Rational Rose Enterprise.

#### **4.11. Conclusiones.**

- En este Capítulo para la implementación del Tutorial de IA trabajamos con PHP 5.2.3, el cual es uno de los lenguajes de programación Web más usados hoy en día, el cual nos brinda muchas facilidades en la programación. Conocimos sus principales ventajas. Como servidor Web aprendimos las características del Apache y trabajamos con la herramienta de diseño de la aplicación Dreamweaver 8 con lo cual diseñamos el Sitio y la Interfaz de Usuario.
- Aprendimos las facilidades brindadas por MySQL en la implementación de la Base de Datos, también utilizando la Herramienta Rational Rose representamos los Diagramas de los Casos de Uso y representamos el Diagrama de Navegación del Tutorial de Inteligencia Artificial. Por último con la información brindada por el Departamento de Informática de los temas de Inteligencia artificial se colocó en el Red Intranet de la Universidad para acceder a este Tutorial.

## CONCLUSIONES

- ✓ Se potenció el estudio de las técnicas, funcionamiento y aplicaciones de la Inteligencia Artificial mediante el apoyo de una aplicación educativa.
- ✓ Se implementó un tutorial de aprendizaje de Inteligencia Artificial que permite apoyar la solución de la problemática existente.
- ✓ Con la realización de este trabajo se estableció parámetros de comparación y evaluó las diferentes técnicas de Inteligencia Artificial
- ✓ Se implementó el tutorial en la Red de la Facultad para su explotación por parte de profesores y estudiantes.



## RECOMENDACIONES

- Implementar nuestra aplicación en Centros de Estudios Superiores dadas las múltiples aplicaciones de sus técnicas y principios básicos que ofrece sobre Inteligencia Artificial.
- Publicar en Internet o Intranets el Tutorial de Inteligencia Artificial para que se pueda acceder libremente.
- Estimular a la comunidad académica a continuar investigando sobre el desarrollo de esta prometedora área de Inteligencia Artificial.
- Que se utilice el presente trabajo como material de estudio para los estudiantes de la carrera de Informática de la Universidad de Pinar del Río Hermanos Saíz Montes de Oca.

## REFERENCIAS BIBLIOGRAFICAS

1. **ABCdatos.** Programas y Tutoriales: Inteligencia Artificial Historia y evolución.  
<http://www.abcdatos.com/tutoriales/programacion/inteligenciaartificial.html>  
2008-09-12
2. **Apache Software Foundation.** Apache License versión 2.0, 2008.  
<http://www.apache.org/>  
2008-12-01
3. **Booch, R.** El Lenguaje Unificado de Modelado: Conceptos, 2000.  
pp 50-120.
4. **Borrego, J.** Lógica Computacional. Sevilla-España: Universidad de Sevilla. Dpto. de Ciencias de la Computación e Inteligencia Artificial  
<http://www.cs.us.es/~joaquin/tema-02.pdf>.  
2008-11-11
5. **Cañas, J.** Universidad Técnica Federico Santa María Valparaíso  
<http://www.inf.utfsm.cl/~jcanas/ramos/Fundamentos1/Apuntes/C2cl1.pdf>  
2008-05-12
6. **Carbonell, P.** Control Fuzzy Adaptativo estructuras y estabilidad.  
<http://www3.unileon.es/estylf2002/>  
2008-10-02
7. **Choque, G.** Inteligencia Artificial Perspectivas y realizaciones: Redes Neuronales. La Paz-Bolivia, 2002.
8. **Costa, C.** Algoritmos genéticos básicos ,2002.  
<http://ccp.servidores.net/genetico.html>.  
2008-09-03
9. **Daza, S.** Universidad Militar Nueva Granada Colombia.  
<http://www.monografias.com/trabajos12/redneur/redneur.shtml>  
2008-10-05

10. **FortuneCity**. Representación de Conocimiento. EUA.  
<http://www.fortunecity.com/skyscraper/romrow/207/se/representacion.htm>  
2008-02-06
11. **Fuente, M.** Redes neuronales. Universidad de Valladolid.  
Departamento de Ingeniería en Sistemas y automática.  
[http://www.isa.cie.uva.es/~maria/redes\\_neuronales.ppt](http://www.isa.cie.uva.es/~maria/redes_neuronales.ppt)  
2009-01-05
12. **Galeón, S.** Fundamentos Básicos Algoritmos Genéticos. 1994.  
<http://eddyalfaro.galeon.com/geneticos.html>  
2008-11-05
13. **García, A.** Rincón del Programador, 2001.  
<http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer>  
2009-01-05.
14. **García, J.** España, 2004.  
<http://www.webestilo.com/php/php00.phtml>  
2008-10-18
15. **Gutiérrez, J.** Redes Neuronales. Universidad Católica San Pablo, Perú, 2003.  
[http://www.usp.edu.pe/~jc.gutierrez/IA\\_/IRN.pdf](http://www.usp.edu.pe/~jc.gutierrez/IA_/IRN.pdf)  
2009-01-11
- 16...**Jiménez, A.**, Redes Semánticas y Marcos: Universidad de Sevilla  
Dpto. Ciencias de la Computación e Inteligencia Artificial, 2002.  
<http://www.cs.us.es/cursos/ia2-2002/temas/tema-4.pdf>  
2008-12-16
17. **Meseguer, P.** Búsqueda Heurística, España, 2006.  
<http://www2.iii.csic.es/~pedro/busqueda1-introduccion.pdf>  
2008-08-11
18. **Morales, E.** Representación del Conocimiento  
<http://ccc.inaoep.mx/~emorales/Cursos/RdeC/principal.html>  
2008-12-12
19. **Moreno, A.** Universidad de Málaga ,2000.  
<http://elies.rediris.es/elies9/4-3-2.htm>  
2008-12-20

20. **Muñoz, M.** Algoritmos Genéticos. 2005  
<http://taylor.us.es/componentes/miguelangel/algoritmosgeneticos.pdf>  
2009-01-07
21. **Pacheco, A.** Representación del Conocimiento. República del Saber, México ,2007.  
[http://expo.itchihuahua.edu.mx/view.php?f=prog\\_48](http://expo.itchihuahua.edu.mx/view.php?f=prog_48)  
2009-01-22
22. **Peraza, F.** Inteligencia Artificial.  
<http://www.monografias.com/trabajos10/trabajos/trabajos.shtml>  
2009-01-22
23. **Pérez, J.** El Servidor Web Apache,2004.  
<http://www.pache.org/docs/2004.>  
2008-09-18
24. **Programación en castellano.** Tutorial de My Sql.  
[http://www.programacion.com/tutorial/mysql\\_basico/](http://www.programacion.com/tutorial/mysql_basico/)  
2009-02-12
25. **Rodríguez, G.** Algoritmos Genéticos y Computación Evolutiva  
<http://the-geek.org/docs/algen/>  
2009-02-12
26. **Soluciones Redes Neuronales Artificiales,** 2001.  
<http://electronica.com.mx/neural/>  
2009-02-02
27. **Tolmos, P.** Algoritmos genéticos, 2004.  
<http://www.uv.es/asepuma/X/J24C.pdf>  
2008-05-02
28. **TREC,** Soluciones Redes Neuronales Artificiales, 2001.  
<http://electronica.com.mx/neural/>  
2008-07-22
29. **.Universidad de las Américas.** Fundamentos de Lógica Difusa,  
Puebla, 1999.  
[http://www.itq.edu.mx/vidatec/espacio/aiee/fuzzy.ppt.](http://www.itq.edu.mx/vidatec/espacio/aiee/fuzzy.ppt)  
2009-11-18.

30. **Vega, A.** Tutorial de Redes Neuronales: Programa de Profesorado  
Madrid –España  
<http://www.gc.ssr.upm.es/inves/neural/ann2/concepts/structnn.htm>  
2008-07-28
  
31. **Warianos.** Curso básico de Sistemas Expertos,  
<http://www.warianos.org/foros/showthread.php?t=158394>  
2008-09-11
  
32. **Wikimedia.** Wikimedia Foundation. Encyclopedía libre, 2007.  
[http://es.wikipedia.org/wiki/Historia\\_de\\_la\\_inteligencia\\_artificial](http://es.wikipedia.org/wiki/Historia_de_la_inteligencia_artificial)  
2008-06-19

## BIBLIOGRAFÍA

33. **Acedo, J.** Instrumentación y control básico de procesos: Lógica Difusa  
Editorial Universal. España, 2005.
34. **Fit web.** Universidad de Pinar del Río: Web Facultad Informática y Telecomunicaciones, 2008.  
<http://fitweb.upr.edu.cu/>  
2008- 26-12.
35. **Gómez, F.** Sistemas difusos jerárquicos para modelado y control.  
TESIS DE MAESTRO DE CONTROL AUTOMÁTICO México, D.F,  
2005.
36. **Plan D.** Universidad Pinar Río. Facultad de Informática y Telecomunicaciones: Planes de estudio.2007-2008.
37. **Torres, L.** Representación de Conocimiento. Universidad de Colombia. Departamento de Informática, 2001.

## **TRABAJOS DE DIPLOMA CONSULTADOS:**

38. **Córdoba, C.** "Control Neuro Difuso de un Móvil Autónomo".Universidad del Cauca, Trabajo de grado: FIET -, 2003.
39. **Chaguara, J.** "Control Neuro Difuso Aplicado a una Torre" UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS. Lima-Perú, 2005.
40. **Díaz, P,** Aplicación para la gestión de Publicaciones Científicas: "Web de publicaciones científicas". Universidad de Pinar del Río "Hermanos Saíz Montes de Oca" Tesis (en opción al título de Ingeniero en Informática) Junio, 2007.
41. **Galindo, J.** CONJUNTOS Y SISTEMAS DIFUSOS Universidad de Málaga E.T.S.I. Lógica difusa sql. 2000.
42. **Gómez, F.** Sistemas difusos jerárquicos para modelado y control. TESIS DE MAESTRO DE CONTROL AUTOMÁTICO México, D.F. Centro de Investigación de estudios avanzados del Instituto Politécnico Nacional. Octubre de 2005.
43. **López, C.** "Aplicación de la lógica Difusa en Robótica" Programa de Doctorado: Diseño, Análisis y Aplicaciones de Sistemas Inteligentes. Curso: "Planificación y Robótica "LOG DIFUSA.2005.
44. **Vallejo, E.** HERRAMIENTAS CAD PARA DISEÑO DE CONTROLADORES INTELIGENTES EN ROBÓTICA MÓVIL. Universidad del Norte, Barranquilla, Colombia.2002.

## Índice de Tablas

<b>Tablas</b>	<b>Pag</b>
Tabla I.1 Informática clásica vs Inteligencia Artificial	15
Tabla III.1 Modelos y Campos de Aplicación RNA	51
Tabla III.2 Términos de Algoritmos Genéticos	57
Tabla IV.1 Actores del Tutorial de Inteligencia Artificial.	78
Tabla IV.2 Casos de Uso del Sistema	79
Tabla IV.3 Caso de Uso Iniciar Tutorial.	80
Tabla IV.4 Caso de Uso Autenticarse	81
Tabla IV.5 Caso de Uso Consultar Información sobre I.A.	82
Tabla IV.6 Caso de Uso Gestionar Información Adicional	83
Tabla IV.7 Caso de Uso Realizar Evaluación.	85
Tabla IV.8 Caso de Uso Registro Evaluaciones Realizadas.	86
Tabla IV.9 Caso de Uso Administrar Ejercicios Evaluativos.	87
Tabla IV.10 Caso de Uso Gestionar Cuentas de Usuario.	89
Tabla IV.19 Ficheros Internos	98



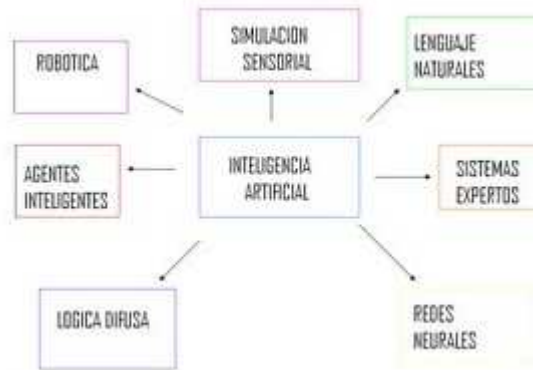
## Índice de Figuras

<b>Figuras</b>	<b>Pag</b>
Figura II.1 Redes Semánticas Básicas	36
Figura II.2 Distancia Semántica	37
Figura III.1 Variable estatura Lógica clásica y (b) lógica difusa.	44
Figura III.2 Esquema general del control difuso.	46
Figura III.3 Estructura Red Neuronal Artificial	49
Figura III.4 Red Neuronal Monocapa	52
Figura III.5 Red Neuronal Multicapa	53
Figura III.6 Red Neuronal Realimentada	53
Figura III.7 Diagrama funcional de un algoritmo genético.	58
Figura III.8 Alternativas de Búsqueda.	62
Figura III.9 Un ejemplo de espacio de búsqueda	65
Figura III.10 Búsqueda Primero a lo ancho.	66
Figura III.11 Búsqueda Primero en Profundidad	67
Figura IV.1 Diagrama de Casos de Uso del Sistema.	79
Figura IV.2 Figura de Iniciar Tutorial (IP).	80
Figura IV.3 Figura de la Interfaz Autenticarse.	81
Figura IV.4 Interfaz Consultar Información sobre I.A. (CIIA).	82
Figura IV.5 Caso de Uso Consultar Información sobre I.A.	83
Figura IV. 6 Gestión de Documentos Adicionales.	84
Figura IV.7 Caso Uso Gestionar Ejercicios Evaluativos.	84
Figura IV.8 Interfaz Realizar Evaluación (RE).	85
Figura IV.9 Interfaz Mostrar Registro de Evaluaciones Realizadas	87
Figura IV.10 Interfaz Administrar Pregunta (AP).	88
Figura IV.11 Interfaz Administrar Respuesta (AR).	88
Figura IV.12 Interfaz Gestionar Cuenta Usuario (GCU).	90
Figura IV.13 Interfaz PHP MyAdmin.	92
Figura IV.14 Base de datos del Tutorial de Inteligencia Artificial	92
Figura IV.15 Mapa de Navegación del Tutorial IA.	94
Figura IV.16 Modelo de despliegue.	95
Figura IV.17 Entradas externas	96
Figura IV.17 Salidas externas	97

## ANEXOS I.2 Paradigmas de Inteligencia Artificial



## Anexo III. 1 Técnicas de Inteligencia Artificial



## Anexo IV. 1 Interfaz de Usuario.

The screenshot shows a Windows Internet Explorer browser window displaying a website. The address bar shows the URL: `http://localhost/iasite/Documentacion/Tema1/Inteligencia.php`. The browser's menu bar includes 'Archivo', 'Edición', 'Ver', 'Favoritos', 'Herramientas', and 'Ayuda'. The toolbar contains various icons for search, anti-spy, update, and email. The website header features a blue banner with the title 'Sitio WEB para la asignatura de Inteligencia Artificial' and two images of a human head with a glowing brain. Below the banner, the user is logged in as 'jose' (Profesor) on '19 de Enero de 2009'. The main content area is divided into three columns:

- Left Column (Table of Contents):**
  - Tema I "Surgimiento y Desarrollo"**
    - Inteligencia
    - Conceptos
    - Relacion con otras ciencias
    - Paradigmas de IA
    - Tendencias de la IA
  - Tema II "Programacion Logica"**
    - Programacion descriptiva
    - Prolog
    - Caracteristicas de Prolog
    - Tipos de datos
    - Listas en Prolog
    - El predicado padre
    - El predicado miembro
    - Estructuras
    - Selectores
    - Control de ejecucion
    - Tipos de Lazos
  - Tema III "Lenguaje natural"**
    - Definicion
    - Lenguaje formal
- Middle Column (Main Content):**
  - Inteligencia**

En primer lugar, revisemos algunas definiciones generales de inteligencia, antes de intentar definir inteligencia artificial. La definición de inteligencia posee diversas aseercciones y actualmente no existe una definición universalmente aceptada.

Inteligencia es la aptitud de crear relaciones. Esta creación puede darse de manera puramente sensorial, como en la inteligencia animal; también puede darse de manera intelectual, como en el ser humano, que pone en juego el lenguaje y los conceptos. También se la puede conceptualizar como la habilidad para adquirir, comprender y aplicar conocimiento; o como la aptitud para recordar, pensar y razonar.
  - Administración**

La Inteligencia puede ser analizada como un Concepto Global. Por ejemplo, algunas personas la relacionan con: "la habilidad para aprender".

Binet y Simon argumentaron que la Inteligencia está formada por: el juicio; el sentido práctico; la iniciativa; la facultad de adaptarse por sí mismo a las nuevas circunstancias y otros factores. Los enfoques para medir y definir la inteligencia están asociados a los mecanismos de aprendizaje y a los resultados del aprendizaje.

El diccionario enciclopédico SALVAT define inteligencia como: "Facultad intelectual con que se captan y forman ideas y se crean relaciones"
  - Perfil de Usuario**

jose  
Jose Salao

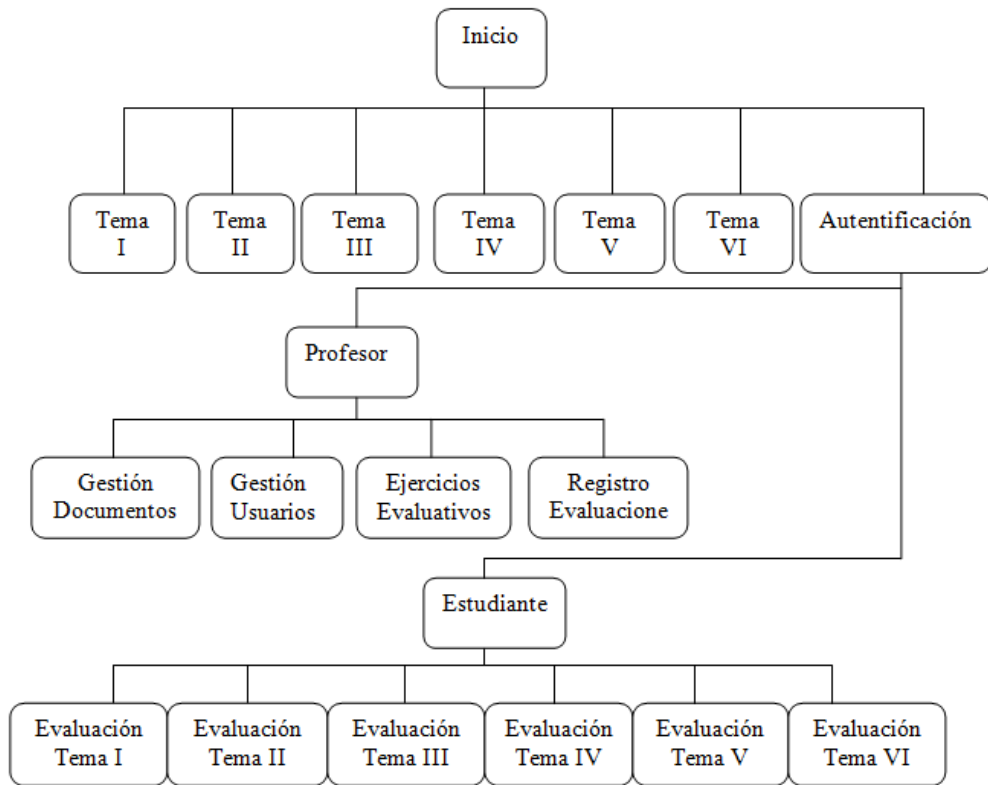
Cambiar Contraseña  
Desconectar
- Right Column (Administration):**
  - Administración**
    - Gestion de usuarios
    - Ejercicios evaluativos
    - Registro de evaluaciones
    - Gestión de documentos

The browser status bar at the bottom shows 'Intranet local | Modo protegido: activado' and a zoom level of '100%'.

## Anexo IV.2: Modelo Lógico de los Datos del Tutorial de I.A.

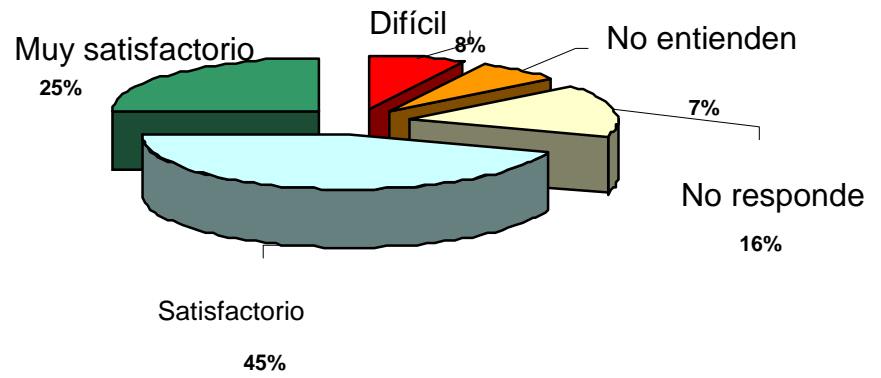
Nº	Tabla	Llave Primaria	Llave Extranjera	Campos	Tipo de Dato
1	Usuario	Id_US		Id_US Nombre Usuario Contraseña IDTipo	Int Varchar Varchar Varchar Int
2	TipoUsuario	Id_Tipo		Id_Tipo Tipo IDP	Int Varchar Int
3	Reg PregResp	IdRPR		IdRPR IdEst IdResp Correcta Fecha	Int Int Int Int Varchar
4	Preguntas	Id_Pregunta		Id_Pregunta Id_Tema Texto Tipo	Int Int Varchar Int
5	Respuesta	Id_Respuesta Id_Pregunta		Id_Respuesta Id_Pregunta Texto Correcta	Int Int Varchar Int
6	Doc	IdDoc		IdDoc Nombre Ruta	Int Text Text

### Anexo IV.3: Mapa de Navegación



#### Anexo IV.4: Pruebas del Sistema

### Prueba Sistema en la Intranet de la Universidad Pinar Rio



**Total: 50**