

Universidad de las Ciencias Informáticas (UCI)

Tesis de Maestría

Título: Sistema Inteligente de
Soporte a la Toma de Decisiones.

Autora: Ing. Kathrin Rodríguez Llanes

Tutores: **Liesner Acevedo Martínez**
Iván Pérez Mallea



"Creo firmemente que la gran batalla se librar  en el campo de las ideas y no en el de las armas, aunque sin renunciar a su empleo en casos como el de nuestro pa s u otro en similares circunstancias si se nos impone una guerra, porque cada fuerza, cada arma, cada estrategia y cada t ctica tiene su ant tesis surgida de la inteligencia y la conciencia inagotables de los que luchan por una causa justa"

Fidel Castro Ruz

INDICE

INTRODUCCIÓN.....	5
CAPITULO 1. FUNDAMENTACIÓN TEÓRICA. ESTADO DEL ARTE.....	12
1.1 INTRODUCCIÓN.....	12
1.2 INTELIGENCIA ARTIFICIAL	12
1.2.1. DEFINICIONES DE INTELIGENCIA ARTIFICIAL	12
1.2.2. HISTORIA DE LA INTELIGENCIA ARTIFICIAL	15
1.2.3. DEFINICION DE SISTEMAS INTELIGENTES.....	15
1.2.4. HISTORIA DE LOS SISTEMA INTELIGENTES	15
1.2.5. CÓMO SE CONSTRUYEN LOS SISTEMAS INTELIGENTES	17
1.3 TECNICAS DE INTELIGENCIA ARTIFICIAL.	17
1.3.1 SISTEMAS BASADOS EN CONOCIMIENTO	18
1.4 RAZONAMIENTO BASADO EN CASOS.....	19
1.4.1 DEFINICIONES DEL RAZONAMIENTO BASADO EN CASOS.	
CARACTERISTICAS	19
1.4.2 ESTRUCTURA DE LOS SISTEMAS DE RAZONAMIENTO BASADO	
EN CASOS.....	21
1.4.3 CLASIFICACION DE SISTEMAS CBR.....	28
1.4.4 IMPORTANCIA DEL RAZONAMIENTO BASADO EN CASOS.....	31
1.5 TOMA DE DECISIONES EN LA ESFERA MILITAR	35
1.5.1 TOMA DE DECISIONES DURANTE EL DESARROLLO DE LAS ACCIONES	
COMBATIVAS.....	35
1.5.2 INFLUENCIA DE LOS SISTEMAS INTELIGENTES PARA LA TOMA DE DECISIONES	
EN LA ESFERA MILITAR.....	36
1.5.3 SISTEMAS INTELIGENTES DE APOYO A LA TOMA DE DECISIONES	
37	
1.6 Conclusiones Parciales	43
CAPITULO 2: MODELADO ANALÍTICO DE LA SOLUCIÓN	44
2.1 INTRODUCCION	44
2.2 PLAT AFORMA DE DESARROLLO. HERRAMIENTAS Y TECNOLOGÍAS .	44
2.2.1 LENGUAJE DE MODELADO	45
2.2.2 HERRAMIENTA CASE	46
2.2.3 METODOLOGÍA DE DESARROLLO.....	46
2.2.4 PROGRAMACIÓN DEL LADO DEL CLIENTE.....	47

2.2.5	PROGRAMACION DEL LADO DEL SERVIDOR	49
2.2.6	BASE DE DATOS	53
2.2.7	APLICACIONES, MÓDULOS Y COMPONENTES DE SAUXE 1.5.4 BETA 54	
✓	Acaxia – 2.5.2 Beta	54
✓	SIDEC – 2.0.5 Beta	54
2.3	MODELADO DE LA SOLUCIÓN	55
2.3.1	ANALISIS DEL ENTORNO DEL PROBLEMA	55
2.3.2	MODELO CONCEPTUAL	55
2.3.3	REQUERIMIENTOS FUNCIONALES	56
2.3.4	FORMULAS Y ECUACIONES	58
2.3.5	REQUERIMIENTOS NO FUNCIONALES	60
2.4	DISEÑO DE LA SOLUCIÓN	63
2.4.1	DISEÑO ARQUITECTONICO	63
2.4.1.1	PATRÓN ARQUITECTÓNICO MODELO VISTA CONTROLADOR (MVC) 63	
2.4.1.2	PATRONES DE DISEÑO	66
2.4.1.3	DIAGRAMA DE COMPONENTES	67
2.4.1.4	MATRIZ DE INTEGRACION DE COMPONENTES	68
2.4.2	DISEÑO DE LOS DATOS	69
2.4.2.1	DIAGRAMA DE CLASES DE LA BASE DE DATOS	69
2.4.3	DISEÑO DE LOS PROCEDIMIENTOS	71
2.4.3.1	DIAGRAMA DE CLASES DEL DISEÑO WEB.	71
2.4.3.2	DIAGRAMA DE SECUENCIA	72
2.5	DISEÑO DE LAS INTERFACES	73
	<i>Fig.26: Inferir Decisión</i>	76
2.6	Conclusiones Parciales	77
CAPITULO 3: VALIDACION DE LA SOLUCION		77
3.1	Introducción	77
3.2	Prueba contra bases de datos	77
	El coeficiente de competencia para cada experto se muestra en la tabla siguiente	81
3.4	Análisis De Los Resultados	81
3.5	Conclusiones Parciales	81

RESUMEN

Tomar decisiones de manera general es un proceso complejo, del cual depende la preservación de los recursos materiales y humanos de la entidad a la que pertenecemos, así como el cumplimiento de las metas de la misma. Pero cuando las situaciones son complejas, la toma de decisiones, se convierte en un proceso engorroso, debido a la gran cantidad de aspectos a tener en cuenta para hacer la selección más adecuada. Para llevarlo a cabo eficientemente los directivos deben consultar un volumen de información importante, que por lo general, se encuentra disperso y en diferentes formatos. De ahí que surja la necesidad de desarrollar un sistema informático que aglomere toda esta información y sea capaz, utilizando potentes motores de búsqueda y algoritmos de procesamiento, de mejorar y agilizar el proceso de toma de decisiones.

Teniendo en cuenta lo anterior, se propone un producto software desarrollado completamente en software libre, que constituye una solución genérica, reutilizable y multiplataforma, que utilizando técnicas de inteligencia artificial, específicamente el razonamiento basado en casos, da la posibilidad a los dirigentes de cualquier institución u organismo de tomar una decisión correcta en un tiempo mínimo, ante una situación dada. Con una arquitectura cliente servidor, este sistema, se consume desde la web como un servicio y puede ser perfectamente integrable lo mismo con un sistema de gestión hasta con un sistema de información geográfica que apoye también el proceso a informatizar.

Palabras Claves: Inteligencia Artificial, Proceso de Toma de decisiones, Razonamiento Basado en Casos, FAR.

INTRODUCCIÓN

Hoy en día los sistemas informáticos han ido ganando terreno que ha favorecido el éxito de los diferentes sectores económicos, científicos, y militar. A diario las organizaciones buscan implementar diversas herramientas o estrategias que les permitan de alguna u otra forma el logro o cumplimiento de sus objetivos. Tomar decisiones es súper importante para lograr esos objetivos, pero es, sin duda alguna, un proceso complejo que conlleva responsabilidades y riesgos que hay que asumir, por lo que es necesario contar con tecnología de punta y estar a la vanguardia para adaptar las nuevas técnicas que vayan surgiendo. Actualmente las empresas grandes cuentan con tecnologías y sistemas de punta que les permiten dar seguimiento a las principales operaciones de cada entidad tales como: planeación, organización, dirección y control, ya que la adecuada administración de las mismas conlleva a poder tomar mejores decisiones que benefician a la empresa y a los objetivos de la esta. De

esta forma la toma de decisión se realiza de manera más sencilla pues les ahorra tiempo y les permite evaluar distintos escenarios antes de tomar la mejor decisión de acuerdo a sus necesidades.

La toma de decisiones ante situaciones complejas a menudo supera nuestras capacidades cognitivas debido a la cantidad de variables involucradas y a sus sutiles interdependencias. También es conocido que el juicio intuitivo humano y su toma de decisiones están lejos de ser óptimo, viéndose deteriorado con la complejidad y el estrés. Debido a la importancia de la toma de decisiones en muchas situaciones, ofrecer una ayuda que mitigue las deficiencias anteriores constituye una de las líneas principales de la ciencia a lo largo de la historia.

Tomar una decisión en una guerra o en la esfera militar de modo general, es un tema bien delicado, puesto que de esta depende la vida de seres humanos que es lo más importante y la preservación de valiosos recursos materiales. Generalmente esas decisiones las toman los jefes correspondientes a cada nivel de mando.

Para tomar una decisión los jefes militares deben tener en cuenta muchos aspectos, que definirán el curso de los acontecimientos; mostrando un comportamiento inteligente, y siendo capaces de determinar soluciones correctas a situaciones complejas de forma rápida. Para ello deben consultar grandes volúmenes de información la cual se encuentra dispersa en diversas fuentes y en diferentes formatos, ya sea en mapas, disposiciones, libros de estrategia o de forma digital, en bases de datos, manuales, etc por lo que el proceso de toma de decisión se torna complejo e implica mucho tiempo y en esta esfera el tiempo es un factor decisivo.

El jefe militar debe tener en cuenta ante una determinada situación: las condiciones naturales que lo rodean, el objetivo de la unidad que dirige o asesora, los recursos (y sus características) y personas de los que dispone, entre otros muchos aspectos para poder plantear misiones a sus subordinados y de esta manera enfrentar la situación en la que se encuentra.

Por otra parte el jefe debe manejar también un grupo de factores que se desencadenan una vez tomada la decisión como por ejemplo cuales serán los efectos futuros, es decir en qué medida los compromisos relacionados con la decisión afectarán el futuro; la reversibilidad, que se refiere a la velocidad con que la decisión puede revertirse y la dificultad que implica hacer ese cambio, y el impacto que provocará tomar una determinada decisión.

Las etapas del proceso de toma de decisiones en las FAR son 4: apreciación, esclarecimiento, toma de decisión y planteamiento de misiones, que hoy por hoy, no se realizan de forma automatizada, sino de manera presencial a través de reuniones y contactos entre los jefes de especialidades y otros oficiales de las fuerza Armadas. El

conocimiento que se extrae como resultado de estos intercambios, puede guardarse o no en formato digital, y es lo que conforma las capas de información, las cuales son procesadas por el jefe y a partir de ahí se procede a tomar la decisión. Gran parte de esta información que el jefe manipula se encuentra almacenada digitalmente, pero el análisis y procesamiento de la misma se hace de manera independiente y manual y debido precisamente a este gran volumen de información que el jefe debe consultar antes de tomar la decisión, este proceso en las FAR se torna largo y complejo.

De todo lo anteriormente expuesto, se deriva la necesidad e importancia que reviste que los jefes militares tengan la posibilidad de contar con un sistema automatizado que les permita, una vez introducida la situación real en que se encuentran y todo lo se relaciona con esta, que el software sea capaz de brindar posibles decisiones acertadas, en un periodo de tiempo muy corto.

Problema Científico:

El gran volumen de información que manejan los jefes de las unidades militares actualmente y la dispersión de esta, dificulta el proceso de toma de decisiones durante la realización de las acciones combativas.

Objeto de estudio:

Proceso de Toma de decisiones durante las acciones combativas.

Campo de acción:

Inteligencia artificial, específicamente el razonamiento basado en casos.

Objetivo General:

Desarrollar una aplicación informática que utilizando técnicas de inteligencia artificial facilite el proceso de toma de decisiones a los jefes durante el desarrollo de las acciones combativas.

Objetivos Específicos:

1. Realizar el diseño teórico metodológico de la investigación.
2. Modelar de la solución.
3. Implementar el sistema.
4. Poner en explotación experimental el sistema.
5. Evaluar el sistema, a través del cumplimiento de los requisitos funcionales.

Tareas para dar cumplimiento a los objetivos

1. Realización de la Base Conceptual de la Investigación

2. Caracterización del proceso de toma de decisiones que se utiliza actualmente en las FAR.
3. Búsqueda bibliográfica de los sistemas similares que existen en el mundo.
4. Búsqueda bibliográfica sobre las tendencias actuales en Cuba y el mundo de como construir de sistemas de este tipo.
5. Estudio de las principales técnicas que utilizan.
6. Selección de la técnica que más se adecua a las necesidades reales del proyecto.
7. Definición de las tecnologías y herramientas a utilizar.
8. Especificación de la arquitectura a usar.
9. Definición del cronograma de ejecución del sistema.
10. Intercambios con los especialistas funcionales.
11. Realización de la licitación de requisitos.
12. Realización de un análisis minucioso del negocio a informatizar.
13. Diseño de la solución.
14. Implementación de la solución.
15. Realización de pruebas al sistema.
16. Despliegue de la solución.
17. Evaluación de la solución propuesta.

Hipótesis: Si se desarrolla un sistema informático basado en inteligencia artificial que sirva de apoyo a los jefes militares en la toma de decisiones, se facilitará este proceso durante el desarrollo de las acciones combativas.

Métodos teóricos de la Investigación:

Histórico-lógico: Sirvió para definir las tendencias actuales en el desarrollo de sistemas inteligentes que sirven de apoyo a la toma de decisiones en la esfera militar y para determinar las técnicas más usadas en la construcción de los mismos.

Hipotético-Deductivo: Este método permitió obtener la hipótesis y a partir de ella inferir conclusiones en el transcurso de la investigación y hacer formulaciones particulares, las cuales sí son validadas por expertos y reafirman la validez de las formulaciones y de la hipótesis general que se sustenta.

Analítico-sintético: Mediante este método se procesó la información y sirvió para arribar a conclusiones en la investigación, además para determinar las actividades que se incluirán dentro de la guía a proponer.

Criterio y análisis de expertos.

Delphi: Para la selección de los expertos.

Métodos empíricos:

La observación: Planificada y dirigida con el fin de realizar la fundamentación teórica del problema.

La entrevista: Se realiza a especialistas de la academia de las FAR para obtener información del fenómeno tratado.

Tabla 1: Cronograma de Trabajo

Etapas del Proyecto	Tarea	Inicial	Final	Responsable(s)
Estudio del Estado del Arte	Estudio del estado del arte en la aplicación de técnicas de IA en sistemas de apoyo a la toma de decisiones.	28/9/09	30/11/09	Kathrin Rodríguez Llanes
Investigación y estudio para la solución	Selección y estudio de la técnica de IA a utilizar.	1/12/09	8/1/10	Kathrin Rodríguez Llanes
	Estudio de las herramientas y tecnologías informáticas a emplear para dar solución al problema.			Kathrin Rodríguez Llanes
	Análisis y Diseño de la solución	9/1/10	21/2/10	-Alejandro Aparicio -Luis Frómata
Desarrollo del módulo	Implementación del Sistema inteligente de soporte a la toma de decisiones	22/2/10	5/5/10	-Alejandro Aparicio -Luis Frómata
	Validación de la solución	6/5/10	14/6/10	-Alejandro Aparicio -Luis Frómata

Resultados Esperados

Nuestra principal aspiración al concluir el proyecto es contar un sistema informático que agilice y sirva de apoyo al proceso de toma de decisiones en las FAR durante las acciones combativas con el objetivo de preservar vidas humanas y recursos materiales y que sirva además como instrumento didáctico en la Academia para que los militares que allí se entrenan egresen mejor preparados desde el punto de vista táctico, de mando y dirección. De esta forma se estará contribuyendo al logro de la soberanía tecnológica del país y al fortalecimiento de la defensa de nuestro territorio nacional.

Aporte Teórico y Práctico

Los aportes más significativos de este trabajo son: primero contar con una base de conocimientos sólida con carácter multidisciplinario, porque toma variables de todas las especialidades de las FAR (representación explícita del conocimiento de expertos en el tema), resultado de un trabajo de recopilación importante de información que sirva de apoyo a la toma de decisiones de los jefes militares y segundo contar con un software inteligente capaz de proponer al jefe decisiones acertadas en corto tiempo cuando las situaciones son complejas, teniendo en cuenta experiencias pasadas almacenadas en la base de casos.

Tabla 2: Costo de la Investigación

CLIENTE: Academia de las FAR				ORDEN DE TRABAJO: 1			
PRODUCTO: Software				FECHA DE PEDIDO: septiembre 2009			
CANTIDAD: 1				FECHA DE INICIO: 28 septiembre 2010			
FECHA DE ENTREGA: julio 2010							
FECHA DE TERMINACIÓN: junio 2010							
COSTOS DIRECTOS DE MATERIALES						COSTOS IND. DE PRODUCCIÓN	
Fecha	Tipo	Cantidad	UM	Precio	Importe	Tipo	Importe
	DVD-RW	5	u	\$3,00	\$15,00	Depreciación de pc	\$1 220
	Bolígrafos	60	u	\$1,00	\$60,00	Servicios	\$490,00
	Lápices	60	u	\$0,50	\$30,00	Combustibles	\$1 000
	Goma de borrar	60	u	\$0,50	\$30,00	Salarios Indirectos	\$12 000
	Memoria USB	3	u	\$20,00	\$60,00		
	Papel Bond	12	paquetes	\$5,00	\$60,00		
	Tonel de impresora	2	u	\$70,00	\$140,00		
Total					\$395,00	Total	\$14 710
COSTO DE LA MANO DE OBRA						RESUMEN	
Fecha	Cargo	Cantidad	Mes	Salario	Importe		Importe
	Analista	2	9	\$520,00	\$9360	Materiales directos	\$395,00
	Jefe de Proyecto	1	9	\$750,00	\$6750	Mano de obra directa	\$47014,5
	Diseñador	2	9	\$520,00	\$9360	Costo directo	\$47409,5
	Arquitecto	1	9	\$615,00	\$5535	Costos indirectos	\$14 710
	Desarrollado	3	9	\$520,00	\$14040		

	r						
		Tasa	Base		\$31005		
	Vacaciones	9,09 %	\$31005		\$2818,35	Costo total	\$62119,5
	Impuesto Fuerza Trabajo	25%	\$33823, 35		\$8455,83		
	Seguridad Social	14%	\$33823, 35		\$4735,27	Unidades producidas	1
Total					\$47014,5	Costo unitario	\$62119,5

Estructura de la tesis

La tesis está estructurada de la siguiente forma: en el capítulo 1 se realiza una fundamentación teórica del tema en cuestión, vista desde el ámbito nacional e internacional. En el capítulo 2 se desarrolla la modelación analítica de la solución, donde se define la plataforma tecnológica, los algoritmos y procedimientos a utilizar, se realiza el análisis y diseño del sistema y en el capítulo 3 se implementa y valida la solución propuesta.

CAPITULO 1. FUNDAMENTACIÓN TEÓRICA ESTADO DEL ARTE

1.1 INTRODUCCIÓN

En el presente capítulo se pretende abordar ideas generales y principales conceptos relacionados con los software que utilizan inteligencia artificial para apoyar la toma de decisiones en la esfera militar, para lo cual se ha dividido el tema en dos subtemas o temas puntuales: la inteligencia artificial y la toma de decisiones en la esfera militar que sirven de apoyo en el desarrollo conjunto de la temática.

Se incluye además un estado del arte del tema tratado, a nivel internacional y nacional donde se reflejan las ideas que en la actualidad han servido de base para la solución del problema que se enfrenta y sobre las que se profundiza más adelante.

1.2 INTELIGENCIA ARTIFICIAL

1.2.1. DEFINICIONES DE INTELIGENCIA ARTIFICIAL

Hoy el término **inteligencia artificial** es un concepto manejado por todos y a pesar de que nos queda mucho terreno por pulir en este sentido y mucho más por investigar, se puede aseverar que se han dado pasos sólidos en el desarrollo de esta rama de la ciencia de la computación. Las definiciones de inteligencia artificial son ya comúnmente aceptadas por la comunidad científica internacional, aparecen en los textos y en las clases y conferencias de los profesionales más avezados en el campo de la informática. Se ha convertido en la actualidad en la meta principal para cualquier empresa productora de software que se dedique a la automatización de tareas que requieran un comportamiento inteligente en cualquier parte del mundo.

Resulta suficientemente clara la definición que ofrece **Farid Fleifel Tapia**.

"La Inteligencia Artificial es la rama de la ciencia de la computación que estudia la resolución de problemas no algorítmicos mediante el uso de cualquier técnica de computación disponible, sin tener en cuenta la forma de razonamiento subyacente a los métodos que se apliquen para lograr esa resolución".

En la IA se puede observar dos enfoques diferentes:

1. La IA concebida como el intento por desarrollar una tecnología capaz de proveer al ordenador capacidades de razonamiento similares a los de la inteligencia humana.
2. La IA en su concepción como investigación relativa a los mecanismos de la inteligencia humana que se emplean en la simulación de validación de teorías.

El primer enfoque se centra en la utilidad y no en el método, los temas claves de este enfoque son la representación y gestión de conocimiento, sus autores más representativos son John McCrrthy y Marvin Minsky.

En el segundo enfoque encontramos que este se orienta a la creación de un sistema artificial capaz de realizar procesos cognitivos humanos haciendo importante ya no la utilidad como el método, los aspectos fundamentales de este enfoque se refieren al aprendizaje y adaptabilidad y sus autores son Newell y Simon de la Carnegie Mellon University.[\[31\]](#)

Para completar esa definición a continuación se relacionan algunas otras no tan formales emitidas por diferentes investigadores de la IA que consideran otros puntos de vista:

- A. Barr y E. A. Feigenbaum, dos de los pioneros de la investigación en IA, definen esta como sigue: "La Inteligencia Artificial es la parte de la Ciencia que se ocupa del diseño

de sistemas de computación inteligentes, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la comprensión del lenguaje, el aprendizaje, el razonamiento, la resolución de problemas, etc.

- “La interesante tarea de lograr que las computadoras piensen... máquinas con mentes, en su amplio sentido literal”. (Haugeland, 1985)
 - “La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades tales como toma de decisiones, resolución de problemas, aprendizaje...”(Bellman, 1978)
 - “El arte de crear máquinas con capacidad de realizar funciones que realizadas por personas requieren inteligencia”. (Kurzweil, 1990)
 - “El estudio de cómo lograr que las computadoras realicen tareas que, por el momento, los humanos hacen mejor”. (Rich y Knight, 1991)
 - “El estudio de las facultades mentales mediante el uso de modelos computacionales”. (Charniak y McDermott, 1985)
 - “El estudio de los cálculos que permiten percibir, razonar y actuar”. (Winston, 1992)
 - “Un campo de estudio que se enfoca en la explicación y emulación de la conducta inteligente en función de procesos computacionales”. (Schalkoff, 1990)
 - “La rama de la computación que se ocupa de la automatización de la conducta inteligente”. (Luger y Stubblefield, 1993)[27]
- Finalmente puede ser aceptada como una definición general, la siguiente: La inteligencia artificial es una rama de la investigación científica dentro de la informática que se encarga de modelar la inteligencia humana a través de sistemas computacionales. Utiliza técnicas complejas para tratar que un ordenador simule el proceso de razonamiento humano. Pretende también que el ordenador sea capaz de modificar su programación en función de su experiencia y que aprenda. [\[24\]](#), [\[25\]](#), [\[26\]](#)

1.2.2. HISTORIA DE LA INTELIGENCIA ARTIFICIAL

Las eras o etapas de la inteligencia artificial.

La inteligencia artificial tuvo su boom a partir de la conferencia efectuada en el Dartmouth College en 1956 (aquí nace la IA), estando presentes científicos como:

- 1.-John McCarthy: quien le dio el nombre a esta nueva área del conocimiento.
- 2.-Marvin Minsky: fundador del laboratorio de IA del MIT.
- 3.-Claude Shannon: de los laboratorios Bell de Estados Unidos.
- 4.-Nathaniel Rochester: de IBM
- 5.-Allen Newell: primer presidente de la A.A.A.I. (Asociación Americana de Inteligencia Artificial)
- 6.-Herbert Simon: premio Nobel de Carnegie Mellon University.

nota: en términos generales las eras por las que ha pasado la IA son las siguientes:

- 1.-El inicio (1956-1965) poniéndose principal énfasis en la implementación de juegos en el computador (ajedrez, damas etc.). Así como en la demostración de teoremas.
- 2.-La etapa oscura (1965-1970) aquí se apoya el entusiasmo por la IA.
- 3.-Etapa del renacimiento de la IA. (1970-1975), iniciado en la universidad de Stanford con el sistema experto médico Mycin (experto en enfermedades infecciosas de la sangre como la meningitis).
- 4.-Etapa de las sociedades. (1975-1980) aquí se identifica la necesidad de trabajar en sociedad con profesionistas en otras áreas del conocimiento (dendral=mycin)
- 5.-Etapa de la comercialización de la IA.[\[30\]](#)

1.2.3. DEFINICION DE SISTEMAS INTELIGENTES

Un sistema inteligente es un programa de computación que reúne características y comportamientos asimilables al de la inteligencia humana o animal.

Es un sistema que aprende durante su existencia (en otras palabras, siente su entorno y aprende, para cada situación que se presenta, cuál es la acción que le permite alcanzar sus objetivos). Actúa continuamente, en forma mental y externa, y al accionar alcanza sus objetivos más frecuentemente que lo que indica la casualidad pura. Consume energía y la utiliza para sus procesos interiores y para actuar.

1.2.4. HISTORIA DE LOS SISTEMAS INTELIGENTES

Si remontamos a épocas históricas, donde los intentos resultaban prematuros en relación a la tecnología disponible, podemos considerar que el camino hacia la construcción de máquinas inteligentes comienza en la Segunda Guerra Mundial (1939-

1945), con el diseño de ordenadores analógicos ideados para controlar cañones antiaéreos o para navegación. Algunos investigadores observaron entonces que existía una semejanza entre el funcionamiento de estos dispositivos de control y los sistemas reguladores de los seres vivos. De este modo, combinando las nuevas teorías sobre la realimentación, los avances de la electrónica de la posguerra y los conocimientos disponibles sobre los sistemas nerviosos de los seres vivos, se construyeron máquinas capaces de responder y de aprender como los animales. Ejemplos clásicos desarrollados en los años cincuenta son las tortugas de W. Grey Walter, que exhibían comportamientos sociales, o **La Bestia** de J. Hopkins, que guiada por un sonar y un ojo fotoeléctrico era capaz de encontrar un enchufe para **alimentarse**.

En el año 1950 Claude Shannon y Alan Turing diseñaron los primeros programas que permitían a un ordenador digital razonar y jugar al ajedrez. A lo largo de los cincuenta se prosiguió trabajando en este sentido.

En 1957 A. Newell, H. Simon y J. Shaw presentaron el Teórico Lógico, el primer programa capaz de razonar sobre temas arbitrarios. Hacia 1960 John McCarthy acuñó el término inteligencia artificial o IA, para definir los métodos algorítmicos capaces de hacer pensar a los ordenadores. En 1965 Marvin Minsky, Newell y Simon habían creado programas de IA que demostraban teoremas de geometría. Aunque importantes, los desarrollos citados (y otros muchos también basados en la manipulación de información simbólica) solamente eran capaces de resolver aquellos problemas para los que habían sido construidos. Sin embargo, los resultados eran tan alentadores que a finales de los sesenta los investigadores llegaron a pensar que en una década se conseguiría construir una máquina realmente inteligente.

El rápido progreso de la IA culminó en los años setenta con la introducción de los **sistemas expertos**, complejos programas de computador en los que se codifica el conocimiento de expertos en una materia muy concreta (concesión de créditos, diagnóstico de enfermedades, etc.), en forma de reglas de decisión. No obstante, un cuarto de siglo más tarde nuestros ordenadores son miles de veces más potentes que los de la época de los pioneros de la IA y, sin embargo, en general, no resultan mucho más inteligentes. El problema radica en que al binomio lógica booleana-máquina Von Neumann sobre el que se asienta la IA, pese a su gran potencia, presenta problemas a la hora de abordar ciertas tareas, como aquellas denominadas del mundo real, donde la información que se presenta es masiva, imprecisa y distorsionada. Para abordar este tipo de tareas, desde hace unos años (década de los ochenta, principalmente) se han vuelto a retomar (o han surgido) una serie de paradigmas de

cómputo alternativos, como las redes neuronales, los sistemas borrosos, algoritmos genéticos o la computación evolutiva, de los cuales los dos primeros quizás sean los más relevantes y empleados. [14]

En definitiva, hoy en día vuelven a coexistir dos corrientes importantes dentro de la búsqueda de la inteligencia que son, más que alternativas, complementarias. Por una parte la IA *convencional*, basada en algoritmos manipuladores de información simbólica y por otra parte, suelen agruparse las redes neuronales, los sistemas borrosos y otras técnicas, que se incluyen en lo que se ha dado en denominar inteligencia computacional, que en cierta medida imitan las construcciones desarrolladas por la naturaleza.

1.2.5. CÓMO SE CONSTRUYEN LOS SISTEMAS INTELIGENTES

Para resolver un determinado problema, en primer lugar debe idearse un algoritmo, que se materializará en la forma de un programa escrito en un lenguaje de programación que el computador entienda (Java, PHP, C++), por último, el microprocesador deberá ejecutar el programa. Con un programa suficientemente largo y detallado, y con una memoria de suficiente capacidad, siguiendo este esquema podría resolverse en principio cualquier problema de cómputo, desde encontrar las soluciones de una ecuación hasta diseñar un edificio, o incluso intentar simular nuestro propio proceso de razonamiento. Precisamente la idea de partida de la inteligencia artificial clásica es intentar dotar de algún tipo de comportamiento inteligente a los computadores.

1.3 TECNICAS DE INTELIGENCIA ARTIFICIAL.

En la actualidad, la Inteligencia artificial se está aplicando a numerosas actividades realizadas por los seres humanos y se destacan entre otras las siguientes líneas de investigación científicas: la robótica, la visión artificial, técnicas de aprendizaje y la gestión del conocimiento. Entre las principales técnicas y campos que emplea la Inteligencia artificial para dar cumplimiento a estas actividades se encuentran: [24]

Técnicas y campos de la Inteligencia Artificial

- Aprendizaje Automático
- Ingeniería del conocimiento
- Lógica difusa
- Redes neuronales artificiales

- Sistemas reactivos
- Sistemas multi-agente
- Sistemas basados en conocimiento
- Sistemas expertos
- Computación evolutiva
- Estrategias evolutivas
- Algoritmos genéticos
- Técnicas de Representación de Conocimiento
 - Redes semánticas
 - Frames
- Visión artificial
- Audición artificial
- Lingüística computacional
- Procesamiento del lenguaje natural
- Minería de datos

De ellas, entre las más significativas podemos encontrar: las redes neuronales, los algoritmos genéticos, la lógica difusa y los **sistemas basados en conocimiento**. De esta última se estará abordando con mayor profundidad a lo largo del trabajo puesto que es la que se ha seleccionado para el desarrollo de la aplicación.

1.3.1 SISTEMAS BASADOS EN CONOCIMIENTO

Los Sistemas Basados en Conocimientos (SBC) son programas que aplican los conceptos de Inteligencia Artificial. Son la pieza comercial y los que más aplicación se les ha dado en Inteligencia Artificial.

La idea básica de estos programas es capturar en un ordenador la experiencia de una persona experta en un área determinada del conocimiento, de tal modo que una persona no experta pueda aprovechar esta información. Es por ello que se crearon los SBC que basándose en algunas reglas de acción (silogismos) y de análisis de posibilidades nos dan una ayuda muy útil en todas las ramas de la acción humana. De este modo se crearon SBC para tareas genéricas: es decir para la monitorización y el diagnóstico, además de los trabajos de simulación de la realidad.

Existen tres tipos de Sistemas Basados en Conocimientos:

- **Basados en reglas:** Aplicando reglas heurísticas apoyadas generalmente en lógica difusa para su evaluación y aplicación.
- **Basados en casos o CBR:** Aplicando el razonamiento basado en casos, donde la solución a un problema similar planteado con anterioridad se adapta al nuevo problema.
- **Basados en redes bayesianas:** Aplicando redes bayesianas, basadas en estadística y el teorema de Bayes.

Estas técnicas pueden combinarse para obtener una solución más adecuada del problema en estudio y poder aplicarse en los sistemas informáticos de soporte a la toma de decisiones. A continuación se describen cada una de estas técnicas.

1.4 RAZONAMIENTO BASADO EN CASOS

1.4.1 DEFINICIONES DEL RAZONAMIENTO BASADO EN CASOS. CARACTERISTICAS

Janet Kolodner, define el Razonamiento Basado en Casos, como una metodología para solucionar problemas, en la cual las soluciones dadas a problemas previos, son utilizadas para resolver nuevos problemas. [\[37\]](#)

Se puede decir que los sistemas basados en casos son un tipo de Sistemas Basados en Conocimiento (SBC) que emplean el Razonamiento Basados en Casos como mecanismo de inferencia. El Razonamiento Basado en Casos es una técnica de Inteligencia Artificial que se basa en la utilización de experiencias previas para resolver nuevos problemas mediante la hipótesis problemas similares tienen soluciones similares.

Dado un problema a resolver, el razonador busca, en una base de datos llamada Base de Casos, problemas similares que anteriormente se hayan resuelto con éxito, llamados casos, y adapta las soluciones para dar una solución al problema actual. Este mecanismo de razonamiento es utilizado por los humanos en múltiples problemas y permite que sea un sistema de fácil comprensión.

El Razonamiento Basado en Casos sugiere un modelo de razonamiento que incorpora los aspectos ya mencionados de resolución de problemas, entendimiento y aprendizaje e integra todo ello en meros procesos de memoria. En resumen, estas son las premisas subyacentes al modelo:

- La referencia a casos pasados es interesante y de gran utilidad para tratar situaciones que vuelven a darse. La referencia a situaciones similares es necesaria a menudo para tratar la complejidad de una nueva situación. Por ello, recordar un caso para

usarlo en un problema futuro (e integrar ambos) es, necesariamente, un proceso de aprendizaje.

- Debido a que las descripciones de los problemas son, a menudo, incompletas, es necesario una etapa de entendimiento o interpretación., ya que no puede llevarse a cabo un razonamiento, una resolución adecuada de una nueva situación, si ésta no se entiende con cierta completitud. Se puede considerar que esta etapa es a la vez un prerrequisito y una parte del ciclo de razonamiento, pues el entendimiento de las situaciones mejora conforme progresa el razonador. No obstante, cualquier forma de razonamiento necesita que la situación sea elaborada con suficiente detalle y representada con suficiente claridad y con el vocabulario apropiado para que el razonador reconozca el conocimiento que necesita (sea conocimiento general o casos) para razonar a partir de él. La práctica demuestra que no suele existir un caso pasado exactamente igual que un caso nuevo. Por ello, es muy usual el tener que adaptar la solución pasada para que se ajuste a la nueva situación.
- El aprendizaje es una consecuencia natural del razonamiento. Si se halla un nuevo procedimiento en el curso de la resolución de un problema complejo y su ejecución resulta positiva, entonces se aprende el nuevo procedimiento para resolver esta nueva clase de situaciones.
- La revisión de la solución propuesta y el análisis de la revisión son dos partes necesarias para completar el ciclo de razonamiento/aprendizaje. Este análisis de la revisión (habitualmente llevada a cabo por un agente externo, léase humano) puede conllevar una reparación de posibles fallos.

Estas premisas sugieren que la calidad de un razonador basado en casos depende de:

- La experiencia que tiene.
- La habilidad para entender situaciones nuevas en términos de experiencias pasadas.
- Su capacidad de adaptación.
- Su capacidad de evaluación y reparación.
- Su habilidad para integrar nuevas experiencias en su memoria adecuadamente.

El CBR involucra toda una metodología con un ciclo de actividades que además de solucionar nuevos problemas nos permita aprender de las buenas soluciones obtenidas por los nuevos problemas:

- **Recuperar:** Dado un problema, se recuperan los casos más similares de la Base de Casos. Un caso es un problema anterior con su solución.
- **Reutilizar:** Extraer la solución del caso seleccionado para utilizarla. Esto puede implicar adaptar la solución a la nueva situación.

- **Revisar:** Se debe analizar si la nueva solución es aceptable y si es necesario revisarla.
- **Retener:** Después de haber aplicado las solución con éxito, se debe almacenar la experiencia como un nuevo caso en la Bases de Casos.

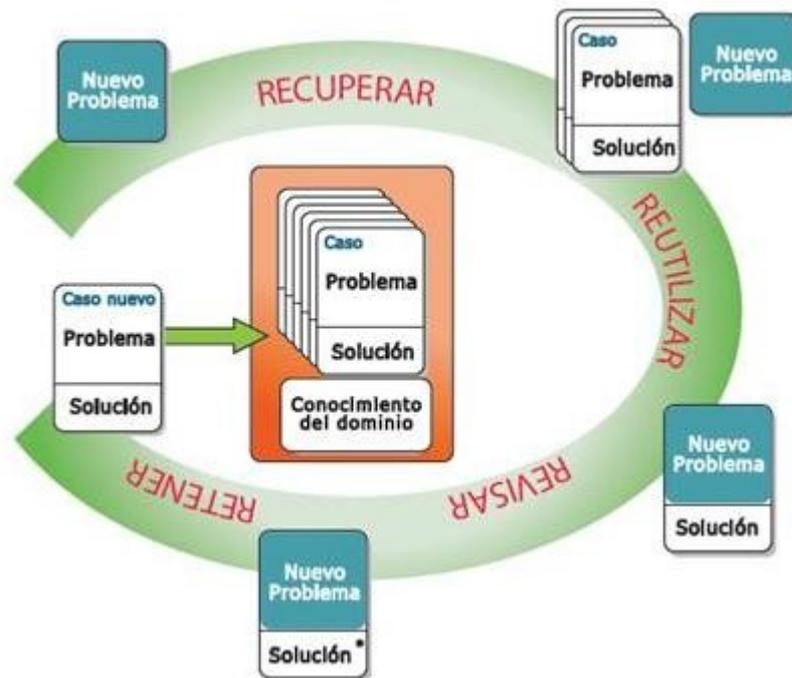


Fig.1: El ciclo del CBR según la UDT-IA

Por lo tanto, para utilizar CBR es conveniente disponer de casos de éxito para diferentes problemas y conocer los diferentes factores que influyen en la solución. Luego será necesario tener un conocimiento sobre el dominio que nos permita evaluar y mejorar las soluciones propuestas.

1.4.2 ESTRUCTURA DE LOS SISTEMAS DE RAZONAMIENTO BASADO EN CASOS

Los sistemas de razonamiento basado en casos constituyen una manera inteligente y flexible de guardar y recuperar información. En cumplimiento de este objetivo incluyen tres componentes principales: **la base de conocimiento** (contiene el conocimiento y las experiencias de los expertos en un determinado dominio), **el motor de inferencia** (es el mecanismo que obtiene las conclusiones de la base de conocimiento mediante procesos de búsqueda) y la **interfaz visual** a través de la cual los expertos introducen el conocimiento por medio de casos.

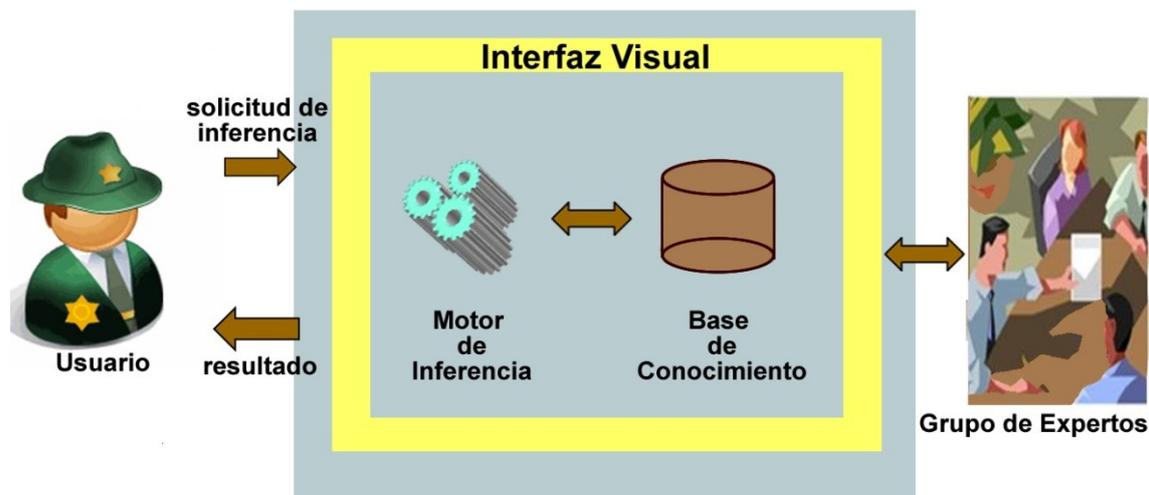


Fig.2: Componentes de un CBR

La base de conocimiento está compuesta por casos. En la terminología CBR, podemos definir un caso como una situación de un problema. Es decir, un caso es la definición completa, clara y precisa de las características de un problema particular que lo distinguen de entre otros problemas, y las acciones que se deben tomar para su corrección. Cada caso está compuesto por un nombre y una lista de características con valores de esas características. De esta manera, una situación previamente experimentada, que ha sido capturada y aprendida de manera que pueda ser reutilizada para resolver futuros problemas, se denomina un caso previo, caso almacenado ó caso guardado. Así, un caso nuevo ó un caso sin resolver no es más que la descripción de un problema nuevo a resolver (donde “resolver” puede ser desde justificar o criticar una solución propuesta, a interpretar el problema, generar un conjunto de soluciones posibles ó generar expectativas de datos observados).

Independientemente de la representación que elijamos para representar un caso siempre debemos tener en cuenta que la información que almacenada en cada uno de ellos debe ser relevante tanto para el propósito del sistema como para asegurar que siempre será elegido el caso más apropiado para solucionar un nuevo problema en un determinado contexto.

El motor de inferencia contempla un conjunto de algoritmos que posibilitan el análisis de la información contenida en la base.

La interfaz visual es lo que posibilita a los expertos, de una forma interactiva y amigable, introducir en la base de casos todo el conocimiento y las experiencias acumuladas sobre el tema en cuestión. También es el componente a través del cual el usuario solicita al razonador que haga la búsqueda que requiere.

Estas partes esenciales en el diseño de un Sistema Basado en Conocimientos se interrelacionan entre sí para obtener las conclusiones necesarias en la resolución del problema en estudio logrando que el sistema diseñado pueda emular el comportamiento del experto en ese dominio específico.

1.4.2.1 DESCRIPCIÓN DE LA BASE DE CASOS

El componente principal de un SRBC, es la base de casos, la cual constituye la memoria del sistema, y está formada por una colección de casos. [\[38\]](#)

En los sistemas CBR la base de casos es la memoria de casos almacenados. A la hora de construir la memoria debemos tener en cuenta los siguientes aspectos:

- La estructura y representación de los casos.
- El modelo de memoria usado para organizar los casos.
- Los índices empleados para identificar cada caso.

Según Janet Kolodner [JKDL1996], un caso es una pieza de conocimiento contextualizado que representa una experiencia, en la que se muestra una lección fundamental para consolidar los objetivos de un razonamiento.

Un caso se puede considerar un registro de un problema. La información que se almacena sobre un problema depende tanto del dominio como del propósito para el que el caso se usa.

En un sistema CBR se incluirá información acerca de la especificación del problema y atributos del medio que describen el entorno del problema. La descripción del problema debe incluir las metas que se deben conseguir para resolver el problema, las restricciones de estas metas, así como las características de la situación del problema y las relaciones entre sus partes. La descripción de la solución, será usada cuando nos encontremos en una situación similar por lo que hay que tener especial cuidado en la información que se almacena. Dependiendo de cómo el sistema razone con los casos, se puede incluir únicamente los hechos que llevan a la solución o información sobre pasos adicionales en el proceso de obtención de la solución, aunque también es posible incluir soluciones alternativas junto con las razones y justificaciones que motivaron la elección de unas a favor del resto, o también soluciones no admisibles rechazadas junto con sus motivos de su rechazo.

Por último es importante incluir una medida del éxito si en la base de casos se han logrado soluciones con diferentes niveles de éxito o fracaso y una medida del resultado obtenido, si se han cumplido las expectativas o no ante el problema.

El conocimiento que almacena un caso es específico, no abstracto, y todo el conocimiento relacionado (conocimiento aplicable en una circunstancia concreta) se encuentra cerca en la base de casos, esto quiere decir que el conocimiento que necesitemos para resolver un problema específico lo encontraremos agrupado en unos pocos casos o incluso en uno.

La elección de la cantidad de casos que se desean almacenar, cómo se van a representar los mismos y si se hacen o no generalizaciones depende, del que diseñe la aplicación.

1.4.2.2 ESTRUCTURA Y REPRESENTACION DE LOS CASOS

Los casos pueden representar distintos tipos de conocimiento y pueden ser almacenados en diferentes formatos. Esto dependerá del tipo de sistema CBR.

En muchas aplicaciones prácticas los casos se representan como dos conjuntos desestructurados de pares *atributo-valor* que representan el problema y las características de la solución.

Sin embargo, es muy difícil decidir exactamente qué representar. Por ejemplo en un sistema CBR para diagnóstico médico, un caso podría representar el historial médico completo de un paciente o solamente una visita. En caso de ser una única visita, el caso podría ser un conjunto de síntomas junto con el diagnóstico y el tratamiento prescrito. En este caso no se considera la evolución del paciente lo cual sería útil para resolver un problema futuro similar. Si el caso representa el historial médico completo del paciente, se puede incorporar información sobre la evolución del paciente desde una visita a la siguiente. Sin embargo este formato dificulta la búsqueda de un conjunto particular de síntomas para obtener un tratamiento.

Una de las ventajas del razonamiento basado en casos es la flexibilidad que ofrece respecto a la representación. Se puede elegir la implementación adecuada dependiendo del tipo de información a representar.

Independientemente de la representación que elijamos siempre debemos tener en cuenta que la información que almacene un caso debe ser relevante tanto para el propósito del sistema como para asegurar que siempre será elegido el caso más apropiado para solucionar un nuevo problema en un determinado contexto.

1.4.2.3 FORMAS DE ORGANIZACIÓN DE LA BASE DE CASOS. MODELOS DE MEMORIA

La base de casos, puede definirse como una base de datos de casos, donde se almacena gran cantidad de información que debe ser extraída de forma eficiente. Para lograrlo se definen diversos tipos de organización de la base de casos, dependiendo de la forma de almacenamiento y tipo de búsqueda [BA2001] [JTAV2003] [RBDG1988].

Entre las formas de organización más comúnmente utilizadas, se destacan:

- Organización plana.
- Organización Jerárquica.

A continuación se detallan las distintas formas de organización de las bases de casos.

➤ **Organización Plana**

Los casos se almacenan secuencialmente en una lista simple, un arreglo o un fichero. Para lograr una recuperación eficiente, se indexan los casos de la base. En este método los índices se eligen para representar los aspectos importantes del caso y la recuperación involucra comparar las características consultadas con cada caso de la base de casos, es decir, la búsqueda se realiza en serie haciendo un recorrido por todos los casos.

Este tipo organización presenta la ventaja de que añadir nuevos casos resulta muy “barato” (rápido y fácil de implementar). No ocurre así con la recuperación de casos, ya que resulta muy lento cuando el número de casos en la base es alto.

➤ **Organización Jerárquica**

a) Redes de características compartidas

Los casos se almacenan en un árbol o en un grafo acíclico directamente. El grafo subdivide el espacio de casos de acuerdo con los atributos que les caracterizan (clustering). Las características comunes ocupan nodos del grafo de donde cuelgan los casos que las comparten. La mayoría de los sistemas tienen alguna especie de umbral relativo al número de casos que debe compartir el valor de una característica, de modo que se justifique su posición en un nodo.

Este procedimiento es computacionalmente más caro, ya que la base requiere un mayor espacio de almacenamiento y su actualización resulta más compleja ya que requiere generalmente modificar una parte de la red. Por otra parte, el diseño y mantenimiento de la red óptima es costoso. Los casos recuperados dependen del contenido de los nodos: la jerarquía establecida debe responder a la importancia de

las características. El algoritmo de búsqueda del caso que mejor se ajusta al de entrada es también sencillo.

b) Redes de discriminación con prioridades

Los tipos de sistema anteriores no resuelven el problema de búsqueda cuando la entrada está incompleta. En las redes de discriminación con prioridades cada nodo contiene una pregunta para la cual los subnodos correspondientes ofrecen respuestas alternativas. Las preguntas más importantes se formulan primero, situándose más arriba en la jerarquía.

Así como las redes de características compartidas, las redes de discriminación subdividen el conjunto de casos y comparten la mayoría de las ventajas y desventajas de esta técnica. La formulación de preguntas puede implementarse más eficientemente que el emparejamiento en cada subnodo. Al mismo tiempo, separar los atributos de sus valores particulares hace más fácil identificar qué atributos han resultado más útiles para la caracterización del caso. Las desventajas que plantean son las mismas que las de las redes de características compartidas: algunos casos significativos pueden ignorarse si la ordenación de las preguntas no es la óptima.

c) Redes de discriminación redundante o modelos de memoria dinámica

Las redes de discriminación redundantes resuelven este problema organizando los casos mediante varias redes de discriminación, cada una con una ordenación diferente de las preguntas. La variedad más común de las redes redundantes incluye en su organización las propiedades de las redes de características compartidas para mantener su tamaño bajo control. Este modelo de memoria dinámica es el más implementado por la mayoría de sistemas CBR.

El modelo de memoria dinámica fue desarrollado a partir de la teoría general de paquetes de organización de memoria, donde la memoria de casos es una estructura jerárquica de episodios generalizados (GE).

La idea básica es organizar los casos que comparten propiedades similares en una estructura más general (el episodio generalizado).

Un GE contiene tres tipos de objetos: normas, casos e índices. Las normas son características comunes a todos los casos indexados bajo un GE. Los casos, que hacen referencia a la situación planteada. Los índices, que son las características que diferencian a los casos de un GE.

La memoria de casos es una red donde cada nodo es un GE (conteniendo las normas), o un nombre de índice, o un valor de índice, o un caso. Cada par índice-valor apunta desde un GE a otro GE o a un caso. Un valor de índice apunta a un único caso o a un único GE. El esquema de indexación es redundante, puesto que existen múltiples caminos hasta un caso concreto o un GE. Cuando un nuevo caso es dado se busca el mejor emparejamiento y el nuevo caso se coloca en la red comenzando por el nodo raíz.

El procedimiento de búsqueda para el almacenamiento de casos es similar al que se utiliza en la recuperación de casos. Cuando una o más características del caso ajustan con una o más características de un GE, el caso es distinguido por el resto de sus características. Normalmente se encuentra el caso que más características tiene en común con el caso de entrada. Durante el almacenamiento de un nuevo caso, cuando una característica del nuevo caso se ajusta a la de un caso existente se crea un nuevo episodio generalizado. Los dos casos se distinguen indexándolos con diferentes índices bajo el GE.

Si durante el almacenamiento de un caso, dos casos (ó dos GE) finalizan bajo el mismo índice, automáticamente se crea un nuevo GE. Por tanto, la memoria de casos es dinámica en el sentido de que las características que comparten dos casos son dinámicamente generalizadas en un GE y los casos son indexados bajo ese GE por sus características diferenciadoras. Para la recuperación de un caso, se busca el GE que tiene más normas en común con la descripción del problema. Una vez encontrado se recorren los índices bajo el GE para encontrar el caso que contiene más características adicionales en común con el problema.

El almacenamiento de un nuevo caso se realiza del mismo modo, con el proceso adicional de la creación dinámica de GE. Puesto que la estructura de índices es una red discriminatoria, un caso (ó un puntero a un caso) es almacenado bajo un índice que le diferencia del resto de casos. Esto puede hacer crecer de forma explosiva el número de índices a medida que aumenta el número de casos, de forma que la mayoría de los sistemas que usan este esquema de indexación ponen algunos límites a la hora de seleccionar índices para casos, como por ejemplo, sólo permitir un pequeño vocabulario de índices.

El principal rol de un episodio generalizado es representar una estructura indexada para el emparejamiento y recuperación de casos. Las propiedades dinámicas de esta organización de memoria podrían ser vistas como un intento de construir una memoria que integra conocimiento de episodios específicos con conocimiento generalizado de los mismos episodios. Por tanto, esta organización del conocimiento es adecuada para

el aprendizaje tanto de conocimiento generalizado como de conocimiento específico y es un modelo admisible, aunque simplificado, del razonamiento y aprendizaje humano.

d) Modelo categorías-ejemplares

La memoria se compone de una red de estructuras y categorías, semánticas, relaciones, casos y punteros índices. Cada caso se asocia a una categoría. Un índice puede apuntar a un caso o a una categoría, pudiendo ser de tres tipos: enlaces de atributos, desde los descriptores de un problema a los casos o a las categorías, punteros de caso desde una categoría hasta sus casos atribuidos, punteros de diferencias desde un caso hasta los casos vecinos más próximos, con pequeñas diferencias entre ambos. La búsqueda de casos se realiza recorriendo las características del problema hasta obtener los casos o categorías con mayor semejanza. La inserción de casos se realiza buscando la categoría o categorías donde debe insertarse.

En conclusión, se asume que los casos tienen dos componentes: la especificación del problema y la solución. Normalmente, la especificación del problema suele consistir en un conjunto de atributos y valores. Los atributos de un caso deben definir el caso de forma única y deben ser suficientes para pronosticar una solución. La representación puede ser una estructura de datos plana o una compleja jerarquía de objetos.

1.4.3 CLASIFICACION DE SISTEMAS CBR

La clasificación de los distintos tipos de sistemas de Razonamiento Basado en Casos se puede basar bien en función de su aplicación, en la representación que realizan de la biblioteca de casos o del grado de intervención humana en el funcionamiento del sistema.

1.4.3.1 EN FUNCION DE SU APLICACIÓN

➤ Tareas de clasificación

Las tareas de clasificación son habituales en el mundo de los negocios e incluso cualquier día de nuestra vida. Se pueden reconocer cuando se necesita emparejar un objeto o evento con otros en una librería en la cual se puede inferir una respuesta.

Algunos ejemplos de tareas de clasificación en el mundo de los negocios son:

¿Qué tipo de casa es esta? (de lujo, chalet, adosado, cabaña, apartamento, etc.)

¿Qué tipo de tratamiento debe darse al paciente? (observación, esteroides, antibióticos, etc.)

¿Hay petróleo bajo esta tierra? (es posible, es imposible, muy probablemente, no es probable)

¿Cuánto tiempo llevará acabar este proyecto? (3 meses, 6 meses, 1 año, etc.)

Si bien alguna clasificación no tiene valores discretos, sí es posible dar un cierto rango. Por ejemplo, una casa de lujo no tiene por qué ser la que vale exactamente 1 millón de Euros, pero sí las que pueden valer entre 1 y 3 millones de Euros.

Habitualmente, las clasificaciones de las preguntas se refieren a resultados, esto es, un resultado es normalmente un atributo del caso y es por lo que se clasifican los casos.

Podemos aplicar CBR fácilmente a problemas de clasificación puesto que pueden consistir en:

- La recuperación de un amplio conjunto de casos similares, por ejemplo aquellos en los que el antibiótico fue el tratamiento.
- Recuperar el mejor ajuste de este conjunto, quizás para sugerir penicilina como antibiótico específico.
- Adaptar la solución, por ejemplo alterando la dosis para diferentes edades o pesos de los pacientes.
- Almacenar el resultado del nuevo caso para un futuro uso.

Como vemos, las tareas de clasificación son fáciles de implementar porque se ajustan al ciclo CBR, los casos tienden a ser más fáciles de representar y recuperar, y los algoritmos de recuperación utilizados en la mayoría de las herramientas CBR son clasificadores.

➤ **Tareas de síntesis**

Estas tareas son comunes en el comercio pero difíciles de implementar. Esto se debe a que es más fácil ajustar un artefacto a un conjunto de artefactos prototípicos que construir un artefacto a partir de una especificación.

Las tareas de clasificación simplemente requieren reconocimiento de las características mientras que las tareas de síntesis requieren colocar las características correctas en el orden y lugar correcto.

Los sistemas de síntesis operan en dominios de diseño o planificación para intentar simplificar el proceso creativo produciendo un diseño o plan que se sabe que es bueno para producir el plan final a partir de él.

Para los diseñadores, esto es más rápido que empezar un diseño desde una hoja en blanco. Se asume que modificar un buen diseño o plan inicial es más fácil que crear uno desde el principio.

En muchas circunstancias esto es cierto, sin embargo hay muchas situaciones en que se debe empezar desde cero sin tener referencia de ningún ejemplo pasado. Por ejemplo, muchos cohetes espaciales se diseñan desde cero para no cometer los mismos errores.

Las razones por las que los sistemas de síntesis son difíciles de construir son:

- La representación de un caso de un plan o diseño es compleja y altamente estructurada con muchas dependencias entre características. Los casos no se almacenan en un medio único y homogéneo, por tanto la recuperación de casos es más difícil.
- Las herramientas CBR tienden a no soportar indexación o recuperación de representaciones de casos altamente estructurados.
- La adaptación es a menudo un requisito clave en las tareas de síntesis.

1.4.3.2 SEGÚN LA REPRESENTACION DE LOS CASOS

Atendiendo a este criterio, podemos dar una clasificación de los sistemas como:

➤ **Razonamiento basado en ejemplo**

Cada concepto se define por extensión, como un conjunto de ejemplares. De esta forma se puede intentar aprender varias definiciones de un mismo concepto en modo ejemplar de definición. Este tipo de sistemas se utilizan en tareas de clasificación.

➤ **Razonamiento basado en instancia**

Es una especificación del anterior con una alta notación sintáctica. Se requiere un alto número de instancias para un correcto funcionamiento correcto. Las instancias suelen ser vectores simples, sobre todo si se utiliza con aprendizaje automático.

➤ **Razonamiento basado en memoria**

Se representa una colección de casos como una memoria, siendo el razonamiento el direccionamiento sobre dicha memoria. Suele utilizarse en procesamiento en paralelo.

➤ **Razonamiento basado en caso**

Cada caso dispone de abundante información sobre la situación que representa, con una organización interna un tanto compleja. Las propuestas deben ser modificables (adaptables).

➤ **Razonamiento basado en analogía**

Es CBR aplicado a resolución de problemas basándose en similitudes con casos de dominio diferentes.

1.4.3.3 SEGÚN EL GRADO DE INTERVENCION HUMANA

Atendiendo al nivel de intervención humana en el sistema CBR se tiene:

➤ **Sistemas CBR de soporte**

Su misión es recuperar situaciones parecidas a la actual con la solución que aporta cada caso. El resultado se muestra al usuario para que actúe en cada caso con la mayor información posible para el problema planteado. Este tipo de sistemas es el utilizado normalmente en sistemas de soporte a la toma de decisiones. La intervención humana es la selección y adaptación de los mejores casos a la situación actual.

➤ **Sistemas CBR textuales**

Su conocimiento se extrae de una colección de documentos de texto realizados por expertos que solucionaron problemas previos. Su misión posterior es la de soporte, siendo un subconjunto de los primeros con problemática propia de extracción de conocimiento de documentos de texto sin formato ni estructura.

➤ **Sistemas CBR interactivos**

Son sistemas que recuperan situaciones para que el usuario acomode su situación actual con alguna de las posibilidades que se le plantean. Estas posibilidades se extraen de los casos recuperados con la interpretación de la situación actual previa. La intervención humana es la interpretación correcta de la situación actual y la selección del mejor caso.

➤ **Sistemas CBR autónomos**

Su funcionamiento es totalmente carente de la intervención humana. Son sistemas inteligentes que toman decisiones propias en el dominio de aplicación.

1.4.4 IMPORTANCIA DEL RAZONAMIENTO BASADO EN CASOS.

En una situación ideal, un sistema inteligente es tal que se comporta en la misma forma que lo haría un experto humano, presentando ciertas ventajas respecto al humano. La potencia de un sistema inteligente se basa más en una gran cantidad de conocimiento que en un formalismo deductivo muy eficaz. La idea que se persigue cuando se construye un Sistema de Razonamiento Basado en Casos (CBR) es la de

automatizar la labor del experto, partiendo en ocasiones de información insuficiente o incompleta.

Teniendo esto en cuenta, se puede aseverar que un sistema de conocimientos basado en casos no es un sistema pensado para reemplazar al experto humano sino un sistema pensado para ayudar al experto humano en la toma de decisiones y además supone una descarga del experto en el trabajo rutinario y, por lo tanto, la reducción de sus problemas.

El uso de Sistemas de Razonamiento Basado en Casos es especialmente recomendado en las siguientes situaciones:

- Cuando los expertos humanos son escasos en determinada materia.
- En situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas.
- Cuando sea muy elevado el volumen de datos que ha de considerarse para obtener una conclusión.
- En situaciones deterministas, en las que las conclusiones se obtienen aplicando un conjunto de reglas dado. [\[34\]](#)

Los métodos de Razonamiento Basado en Casos han sido abordados desde diferentes aspectos, y ha encontrado aplicaciones interesantes, como lo han sido los trabajos de (Schimdt y otros 1995, Evans 1995) en el campo de la medicina. [\[28\]](#)

En nuestro país se han logrado resultados de relativa importancia en la elaboración de software para el Razonamiento Basado en Casos donde estas aplicaciones han estado orientadas al diagnóstico médico. Estos resultados han sido no solamente de interés práctico, sino también en el plano teórico, a lo que ha contribuido el gran volumen de información que se ha registrado en Bases de Casos.

De esta manera, el Razonamiento Basado en Casos promete ser una herramienta poderosa para la solución de problemas disímiles. Es por ello que constituyen en la actualidad un activo campo multidisciplinar, en el que confluyen investigadores procedentes de muy diferentes áreas, como la electrónica, física, matemáticas, ingeniería, biología o psicología, entre otras.

1.4.4.1 VENTAJAS DE SU USO.

Estos sistemas han demostrado ser herramientas muy útiles en gran cantidad de situaciones. Las ventajas o razones que suponen su uso han motivado el enorme crecimiento de este campo. Algunas se exponen a continuación:

- Aglomerar en un mismo lugar el conocimiento de muchos expertos en una materia dada.
- Reducción en la dependencia de personal clave. Esto se debe a que los conocimientos del personal especializado son retenidos durante el proceso de aprendizaje, y están listos para ser utilizados por diferentes personas. Esto es útil cuando la experiencia es escasa o costosa, o bien, cuando los expertos no se encuentran disponibles para la solución de un problema en particular.
- Con la ayuda de un SE personas con poca experiencia pueden resolver problemas que requieren un “conocimiento especializado”.
- Recordar la experiencia previa lo cual es particularmente útil para evitar la repetición de errores que en que se ha incurrido en el pasado, pues es posible alertar al razonador para que tome las medidas que eviten la repetición de errores anteriores.
- Posibilitan diagnosticar fallos con mayor rapidez.
- Facilita el entrenamiento del personal, ya que amplía de forma más rápida los conocimientos de los especialistas y puede ayudar de manera importante, y a costo menor, a la capacitación y adiestramiento del personal sin experiencia.
- Pueden obtener conclusiones y resolver problemas de forma más rápida que los expertos humanos. Son de gran valor en la situación donde el tiempo juega un papel crítico (control de la refrigeración de una central nuclear).
- Debido a la capacidad de los ordenadores de procesar una gran cantidad de información y de realizar un gran número de operaciones en poco tiempo pueden obtener conclusiones realistas en situaciones donde los expertos humanos no pueden.
- Provee gran flexibilidad en el modelado del conocimiento. En contraste con los sistemas basados en modelos, donde muchos problemas no se resuelven por considerarlos fuera de su ámbito ó no se entiende el problema a poco que los datos estén incompletos, los sistemas CBR usan las experiencias pasadas como dominio de conocimiento y dan una solución razonable, previa adaptación, de este tipo de problemas.
- Son sistemas simples de mantener y fácilmente adaptables a dominios donde el conocimiento cambia frecuentemente.
- Permite al razonador proponer soluciones en dominios que no son del todo entendidos por el razonador. En situaciones donde existe poco conocimiento para construir un modelo causal del dominio ó para derivar un conjunto de heurísticas, un razonador

basado en casos sí puede desarrollar un pequeño conjunto de casos que le permitan funcionar.

Hay dominios que son imposibles de entender completamente porque dependen del comportamiento humano impredecible (la economía por ejemplo). Otros porque simplemente no se ha alcanzado el nivel adecuado para su comprensión. Sin embargo, el razonamiento basado en casos permite tomar ciertas premisas y predicciones basándonos en lo que funciona en el pasado.

- Otra ventaja es que el RBC es un enfoque comprensible, ya que se asemeja a la forma en que resolvemos los problemas de la vida real: basamos nuestras decisiones en experiencias (buenas y malas), lo que permite que el sistema vaya aumentando su eficiencia al pasar el tiempo.[\[16\]](#)
- Resolver problemas muy difíciles tan bien o mejor que los expertos humanos.
- Mejora en calidad y eficiencia el proceso de toma de decisiones. Lo anterior implica que las decisiones podrán tomarse de una forma más ágil con el apoyo de un Sistema Experto. Incluso, las decisiones podrán ser consistentes al presentarse situaciones equivalentes. Esto significa que un Sistema Expertos responderá siempre de la misma forma ante las mismas situaciones, lo cual no necesariamente ocurre con las personas.
- Transferencia de la capacidad de decisiones. Un CBR puede facilitar la descentralización de datos en el proceso de la toma de decisiones en aquellos casos que se consideren convenientes. Así, el conocimiento de un experto puede transferirse a varias personas, de tal forma que las decisiones sean tomadas en el nivel más bajo.
- Permite conseguir tareas de planificación más complejas y consistentes.
- Lograr el aprendizaje que tiene lugar a partir de la información almacenada correspondiente a casos que fueron previamente resueltos exitosamente o no.
- Ofrecer soluciones a nuevos casos a partir del análisis de un razonador que interactúa con bases de casos.
- Reducen la tarea de adquisición del conocimiento. Eliminando la tarea de extraer de un modelo o de un conjunto de reglas, como en los sistemas basados en modelos/reglas, la tarea de adquisición de conocimiento basado en un razonador basado en casos consiste en una representación de experiencias/casos almacenados junto a su representación y solución.
- Focalizar el razonamiento hacia partes importantes del problema señalando los rasgos más significativos del asunto analizado.
- Provee un medio de justificación. El razonamiento basado en casos puede dar un caso previo y una solución (con éxito) de forma que puede utilizarse para convencer al usuario o para justificar una solución propuesta al problema actual. Si el usuario

desea una medida de calidad de la solución, el sistema podría cuantificar cuanto éxito tuvo el caso pasado y que grado de similitud hay con el caso actual y el pasado.

- Permite hacer predicciones del posible éxito de una solución propuesta. Cuando la información se almacena teniendo en cuenta el nivel de éxito de las soluciones previas, el razonador basado en casos puede ser capaz de predecir el éxito de una solución propuesta para el problema actual. Obviamente, el razonador tendrá en cuenta no sólo esos niveles de éxito almacenados sino las diferencias entre el caso ó casos recuperados y la situación actual.

1.4.4.2 DESVENTAJAS

Pero no todo son ventajas cuando de Sistemas de Razonamiento Basado en Casos se trata. También, estos software presentan ciertas desventajas como son:

- La tendencia a usar los casos previos ciegamente, confiando en la experiencia previa sin validarla con respecto a la nueva situación.
- Los casos previos pueden predisponer demasiado al razonador a la hora de resolver el nuevo problema.
- Es común, que normalmente no se disponga del conjunto de casos más apropiado para el tratamiento de un problema concreto.
- Confiar en experiencias previas sin validar puede generar soluciones y evaluaciones ineficientes o incorrectas.
- La recuperación de casos inapropiados puede costar un tiempo considerable o llevar a errores muy costosos, que podrían ser evitados por métodos más incrementales.
- Requieren incrementos en la memoria y velocidad de la computadora para mantener una respuesta rápida debido al crecimiento de la base de casos.
- La limitación principal de los sistemas de árboles de decisión y RBC es la necesidad de desarrollar habilidades para la ingeniería del conocimiento y la especialización por dominios.[\[17\]](#)

1.5 TOMA DE DECISIONES EN LA ESFERA MILITAR

1.5.1 TOMA DE DECISIONES DURANTE EL DESARROLLO DE LAS ACCIONES COMBATIVAS.

Como parte de la preparación que se lleva a cabo en el país desde tiempo de paz para enfrentar situaciones de crisis (guerra), se precisan un conjunto de decisiones importantes que definen de cierta forma las medidas que se tomarán ante cualquier ofensiva del enemigo y las acciones que se llevarán a cabo ante este tipo de

situaciones para dar respuesta al adversario. Pero como en realidad la secuencia de las acciones durante un combate es mucho más dinámica y no se puede preconcebir del todo lo que va a suceder, la necesidad de tener centralizada toda la información requerida para tomar una determinada decisión ante un momento de crisis se hace inminente y más necesario aún cuando las tensiones, el estrés, le impiden de cierta forma a los jefes tomar la decisión más correcta, en el menor tiempo posible y teniendo en cuenta todo lo que se necesita, se hace contar con una herramienta informática eficiente que sirva de apoyo para la toma de decisión durante el desarrollo de las acciones combativas.

1.5.2 INFLUENCIA DE LOS SISTEMAS INTELIGENTES PARA LA TOMA DE DECISIONES EN LA ESFERA MILITAR

La esfera militar, debido a sus agudas exigencias, obtiene, procesa y emplea de forma extremadamente cuidadosa la información, en tanto determina necesidades esenciales, busca los datos y la información que requiere, toma decisiones y elabora respuestas precisas de acuerdo con la información y los recursos disponibles.

Los sistemas inteligentes son capaces de acumular enormes volúmenes de información, procesarlos, organizarlos debidamente y proponer qué hacer a los jefes militares ante determinadas situaciones. Sus grandes bases de datos internas permitirán rápidas tomas de decisiones, en diferentes disciplinas para cada escenario posible y un conjunto de prioridades que consideran factores múltiples, efectos deseados y pérdidas o gastos tolerables.

La información contenida en estas bases de conocimiento es sumamente heterogénea: política, económica, geográfica, estratégica y para las operaciones sobre recursos disponibles, escenarios posibles, prioridades, tipificaciones incluyen análisis de perfiles desde diferentes vistas o perspectivas de un objeto (organización), sus objetivos políticos, militares, económicos, comerciales, de dominio; acciones frecuentes, medios de financiamiento, forma de organización, estructura, alcance, dinámica, etcétera.

Los sistemas inteligentes engloban una serie de características fundamentales para cumplir con el objetivo de ofrecer asesoramiento inteligente o tomar una decisión inteligente ante cualquier situación que se presente. En el marco del combate estos sistemas están diseñados para servir de soporte a los complejos análisis que se requieren en el descubrimiento de las tendencias que se siguen durante el proceso de la guerra, para establecer las prioridades de vigilancia, los medios para la observación y el procesamiento de la información, así como proponer planes de enfrentamiento a las contingencias identificadas, según recursos,

fuerzas disponibles y la coordinación necesaria, todo sobre la base de plataformas capaces de conectar efectivamente a la totalidad de los componentes de la red y de captar, registrar, procesar, y elaborar respuestas apropiadas para cada situación puesta a su consideración con el fin de tomar decisiones eficientes y oportunas. Algunos de los aspectos dentro de la planificación y realización del combate del combate para los cuales los sistemas expertos tienen una importancia sin igual son las siguientes:

- Planeación estratégica.
- Análisis de la tendencia del comportamiento del enemigo.
- Interpretación de variables de comportamiento del enemigo.
- Cálculo y asignación de los recursos.
- Análisis de los riesgos.

Se considera que los jefes militares al efectuar la planificación, realización y control de las acciones durante el combate podrían apoyarse en ellos para tomar de decisiones adecuadas y eficientes. Lo anterior con el fin de poder enfrentar los numerosos desafíos que impone el enemigo con la utilización creciente de la tecnología moderna. Estos soportes deben ser capaces de considerar la información cualitativa y a partir de ella diseñar e implementar modelos estadísticos y computacionales que asistan a los decisores en la resolución de los diversos problemas.

En un mundo cada vez más exigente en todos los aspectos de la vida para los seres humanos, donde se exageran aceleradamente contradicciones de todo tipo, se multiplican las redes como respuesta a la necesidad de responder (adaptarse) a un entorno más exigente, coordinar mejor las acciones y explotar de la mejor forma posible los avances científico técnicos determina la fuerza final y de esta puede depender la obtención de ciertas ventajas sobre un competidor o contrincante cercano. Las redes de medios para la obtención, procesamiento de la información, toma de decisiones y elaboración de respuestas para situaciones específicas se impondrán en todos los sectores de la vida militar o civil donde la dinámica de las acciones permita alcanzar una mayor eficacia en el trabajo, donde sea necesario ganar tiempo-como es el caso de redes para la vigilancia de contingencias naturales: ciclones, incendios, terremotos, etc- u obtener ventajas. Hoy, "el que mejor tome la decisión y más rápido" triunfa y mejora su posición.[\[12\]](#)

1.5.3 SISTEMAS INTELIGENTES DE APOYO A LA TOMA DE DECISIONES

¿Qué sistemas digitales basados en Inteligencia Artificial existen en el mundo para apoyar la toma de decisiones en la esfera militar?

Muchos de estos sistemas son para captar lo que está sucediendo en el escenario de guerra e informarlo a un controlador humano.

- El departamento de inteligencia de señales de la Agencia de Seguridad Nacional del gobierno de los Estados Unidos desarrolló un sistema informático llamado Supervisor Neuronal Remoto (RNM), que utilizando redes neuronales y algoritmos genéticos sirve para descifrar y decodificar (leer) los pensamientos humanos y descifrar los potenciales de 30-50 hertz de 5 mili vatios (EMF) que emite el cerebro humano. Es sabido que los humanos estamos rodeados ("aura") de unas frecuencias y amplitudes de ondas EMF características y únicas (frecuencia bioeléctricas) para cada individuo (tal como una huella digital). La detección de la capa de energía electromagnética posibilitaría no solo la vigilancia e interpretación de los pensamientos a través de criptólogos y ordenadores sino también el envío de señales codificadas específicamente al objeto (la víctima) que filtrarían tal cual la radio sintoniza (modula) una estación de radio determinada y que la persona considere que las ideas transmitidas sean propias. [3]

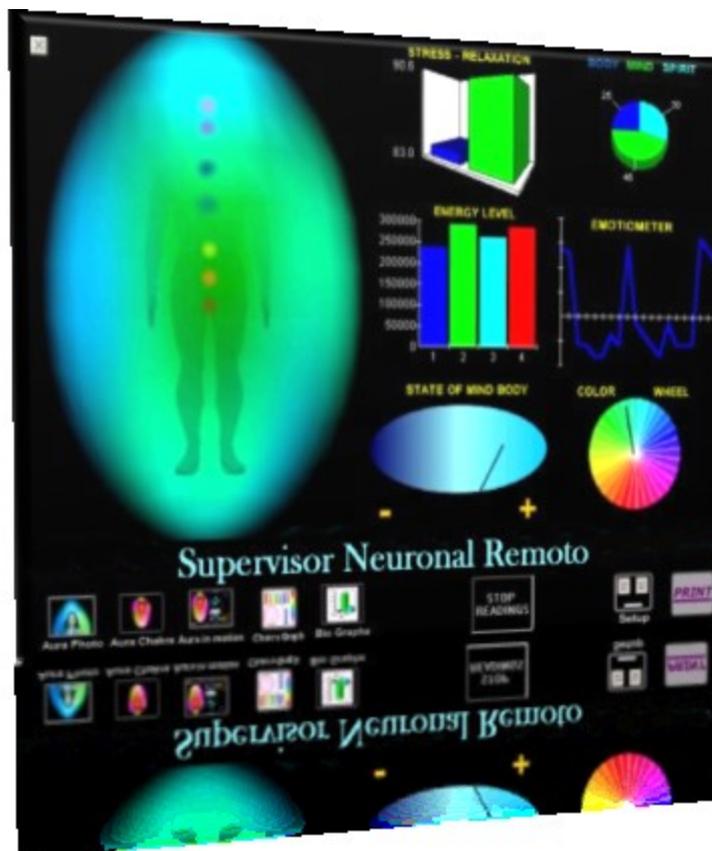


Fig.3: Supervisor Neuronal Remoto

- La aplicación de estructuras y sistemas inteligentes en el ambiente naval tienen marcada influencia táctica, estratégica-operacional y estratégica. La tecnología inteligente mejora el conocimiento informativo...[4]
- El departamento de Defensa de los Estados Unidos desarrolla sistemas de comunicación encubierta, sensores y cámaras que pueden ocultarse con facilidad y que podrían colocarse en regiones remotas y controlarse desde el espacio. [5]
- La compañía norteamericana Watchtower Society desarrolló un conjunto de sistemas robotizados para la defensa denominado **Silver Fox**. Estos sistemas son capaces de enviar información instantánea del escenario de guerra, fijar blancos y regular las rutas y trayectorias de misiles balísticos, usando sistemas georeferenciados de GPS, cámaras fotográficas, haz de luz láser, y video. Otorgando una visión clara del teatro de operaciones. Silver Fox presenta la ventaja de constituir un “enjambre” de naves capaces de entregar “capas” de información (información de varias variables al mismo tiempo), guiadas por un solo controlador humano en tierra.[6]



Fig.4: Sistema robotizado Silver Fox

- En Julio 2000, el Centro de Estudios Estratégicos de la Escuela Superior de Guerra de Argentina desarrolló un simulador táctico inteligente llamado **Batalla Virtual**, para el adiestramiento, apoyo a la toma de decisiones y evaluación de resultados al plantear situaciones tácticas. Tiene como objetivo entrenar a los futuros Oficiales de Estado Mayor en el proceso de apreciación de situaciones, toma de decisiones y el control de las acciones con vista a la conducción de las operaciones en el siglo XXI, con asistencia digital. [7]



Fig.5: Simulador Táctico Inteligente

- En Agosto 2005 el Centro de Estudios Estratégicos de la Escuela Superior de Guerra de Argentina desarrolló un **Sistema Experto para la Modelización de la toma de decisiones**. Procura con la construcción de un motor de inferencias para modelizar el Proceso de Planificación de Comando y tipificar la base de conocimientos del sistema. [7]
- La Escuela Superior de Guerra Aérea de Argentina creó un nuevo juego de guerra que es un verdadero simulador para la toma de decisiones de vuelo y combate; además, desarrolló otro sistema para el planeamiento, comando y control de operaciones aéreas denominado ICC (Integrated Command and Control), y un Juego de Estrategia Competitiva. [21]
- La Academia de guerra del Ejército de Chile desarrolló un Sistema de Gestión y Entrenamiento en red a nivel Nacional para el manejo de situaciones de emergencia y/o catástrofe, llamado **SIGEM** (Simulación para la Gestión y Entrenamiento de Situaciones de Emergencia) que posibilita a las autoridades regionales encargadas de prever, administrar y mitigar situaciones críticas de impacto nacional y regional, tomar decisiones eficientes y oportunas en tiempo real. El sistema tiene un empleo dual de "gestión" ante situaciones reales y "entrenamiento" ante situaciones hipotéticas, posibilitando en ambos casos la implementación del concepto de "lecciones aprendidas", mediante una adecuada arquitectura computacional que implementa "software de agentes inteligentes" para el análisis de decisiones adoptadas (after action review). [8]

- Recientemente la gente de Google publicaban el artículo "[Tour the World: building a web-scale landmark recognition engine](#)", Sistemas Inteligentes al contra-terrorismo utilizando la visión por computador; nueva tecnología para identificar de forma rápida y efectiva más de 50.000 localizaciones en todo el mundo. Permite procesar de forma bastante inteligente imágenes sacadas de satélites o cámaras situadas en zonas públicas, por poner algunos ejemplos, combinando reconocimiento de objetos, reconocimiento de personas, etc.[10]
- En nuestro país se desarrolla el **SIMEM** (Sistema Informático de Mando y Estado Mayor), un proyecto que se realiza de conjunto entre el centro UCI-FAR y la academia de las FAR que tiene como objetivo informatizar todo el proceso de toma de decisiones con sus diferentes etapas y al cual tributará este módulo.



Fig.6: Sistema Informático de Mando y Estado Mayor

- Investigadores de la Universidad de Granada (España) han desarrollado un sistema inteligente para definir el mejor camino dentro de un campo de Batalla Militar, que permitiría a las tropas del Ejército definir el mejor camino dentro de un campo de batalla militar, desarrollado en Delphi 7 a partir de la implementación del algoritmo ACO (Ant Colony Optimization) Optimización Basada en Colonia de Hormigas, considerando que dicho camino lo recorrerá una compañía y que ésta deberá tener en cuenta los criterios de seguridad y rapidez.

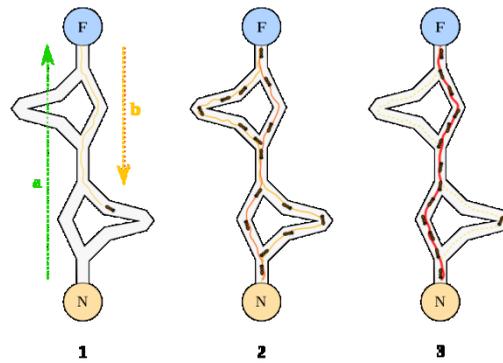


Fig.6: Sistema que utiliza el algoritmo de las colonias de Hormigas

- Sistemas Inteligentes para evaluar conductas de comandantes: JANUS (USA), BBS (USA), SETAC (Chile), Los sistemas de simulación para entrenar a comandantes y sus estados mayores han demostrados ser una efectiva forma de mejorar las coordinaciones, comunicaciones y procesos de toma de decisiones, antes de ejecutar las planificaciones en el terreno con unidades militares reales. Estos sistemas de entrenamiento que permiten evaluar conductas de conductas de comandantes, son conocidos en el contexto internacional como "sistemas constructivos", en donde modelos estocásticos implementados en el computador permiten que dos voluntades representadas por brigadas o divisiones interactúen en un terreno, clima y armamento virtual. [22]

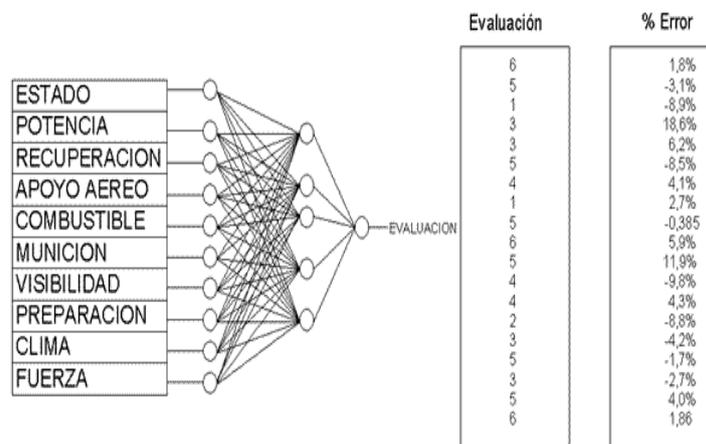


Fig.7: Sistema Inteligente evaluador de conducta

- El Centro de Entrenamiento Táctico Computarizado (CETAC) del Ejército salvadoreño desarrolló un programa informático que simula un conflicto armado internacional donde se enfrentan los ejércitos de dos países imaginarios, uno desplegándose como invasor, y el otro defendiendo el territorio. La complejidad de la situación de combate requiere que los comandantes se entrenen en la toma de decisiones poniendo en práctica los conocimientos teóricos adquiridos en las aulas. El programa ofrece a los

usuarios información sobre recursos, posición de las tropas propias y enemigas y les requiere indicaciones sobre cursos de acción, a la manera de los juegos de simulación comerciales, lo que genera incidentes (virtuales) en los que se producen heridos, prisioneros, víctimas civiles, o destrucción de bienes. [23]



Fig.8: Simulador de Conflicto Armado

- Weka también es un software muy utilizado que basado en IA se utiliza para la inferencia, pero no nos sirve porque no trabaja con atributos multivaluados....

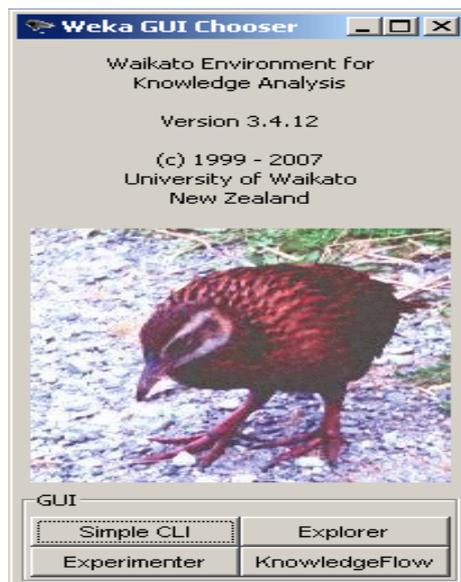


Fig.9: Weka

1.6 Conclusiones Parciales

Una vez analizada la bibliografía relacionada con el tema y encontrados los sistemas informáticos realizados en Cuba y resto del mundo con el propósito de apoyar el proceso de toma de decisiones en la esfera militar, se pudo llegar a la conclusión de que ninguno de los existentes cubre las expectativas que se persiguen en el proyecto, y de los que de algún modo pudieran servir de guía durante la fase de implementación,

no se tienen los detalles suficientes, ya que sus propietarios o sus proveedores no están dispuestos a tratarlos públicamente, por cuestiones de seguridad.

Por las limitaciones de acceso a este tipo de sistemas, en aras de garantizar la soberanía tecnológica que es súper importante en esta rama y por la necesidad de contar con un software que agilice el proceso de toma de decisiones es que se desarrollara un nuevo software que de manera inteligente permita a los jefes militares, una vez introducida la situación en que se encuentran y todos los parámetros relacionados con esta, sea capaz de proponerle una decisión acertada en un corto período de tiempo.

CAPITULO 2: MODELADO ANALÍTICO DE LA SOLUCIÓN

2.1 INTRODUCCION

En el presente capítulo se realiza una descripción detallada de la solución propuesta, que incluye la plataforma de desarrollo seleccionada sobre la cual se analiza, diseña e implementa el software con todas las herramientas y tecnologías a utilizar; el modelado completo de la solución que comprende el análisis de la propuesta, la definición de los requisitos funcionales y no funcionales, entre otros; la vista arquitectónica de sistema, que contempla: los patrones empleados, la línea base definida y los principales componentes que integran la solución, el esbozo de los prototipos funcionales, así como los algoritmos y fórmulas que utiliza el sistema en su funcionamiento interno.

2.2 PLATAFORMA DE DESARROLLO. HERRAMIENTAS Y TECNOLOGÍAS

Una parte indispensable en el desarrollo todo sistema informático es sin dudas definir una plataforma de desarrollo. La plataforma de desarrollo es el entorno común en el cual se desenvuelve la programación de un grupo definido de aplicaciones. Comúnmente se encuentra relacionada de forma directa a un sistema operativo, sin embargo, también es posible encontrarlas ligadas a una familia de lenguajes de programación o a una interfaz de programación de aplicaciones (API).

Para definir dicha plataforma se seleccionan las herramientas y tecnologías a utilizar durante la construcción de las aplicaciones. Para ello, después de un estudio minucioso al respecto y en aras de lograr una homogeneidad, compatibilización e integración entre los productos que se desarrollan en el centro UCID se implementó un marco de trabajo que se nutre de muchas de las funcionalidades y servicios de los frameworks existentes en el mundo, de forma tal que quedaran satisfechas las

necesidades de producción del centro de una manera estándar y optando siempre por la utilización de herramientas libres y tecnología de punta. Como resultado de ello, surge el Marco de Trabajo UCID conocido por Sauxe con una arquitectura en capas (presentación, negocio, servicio, dominio y acceso a datos, datos). Que incluye además el patrón MVC (comunicación entre la capa de presentación y el negocio) y Componentes verticales tales como autenticación, autorización, trazas, entre otros.

2.2.1 LENGUAJE DE MODELADO

Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software.

UML es un lenguaje expresivo, claro y uniforme, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

UML tiene como características:

- ✓ Tecnología orientada a objetos.
- ✓ Viabilidad en la corrección de errores.
- ✓ Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- ✓ Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- ✓ Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura pruebas, versiones, etc.).
- ✓ Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.

Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.

- ✓ Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

UML es independiente del proceso, aunque para utilizarlo óptimamente se debe usar en un proceso que sea dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental. Es importante resaltar que UML es un lenguaje para especificar y no para describir métodos o procesos.

Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema

y para documentar y construir. UML es desde finales de 1997, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle y Rational.

2.2.2 HERRAMIENTA CASE

Durante la etapa de modelación y para diseñar el sistema se utilizó Visual Paradigm, una herramienta CASE de modelado multiplataforma que no se inclina por ninguna metodología específica (se puede modelar cualquier proceso, ya sea RUP, XP, RAID, etc.). Además ofrece un entorno de creación de diagramas para UML, con soporte para los 13 diagramas de la última versión de UML (UML 2.1). Por otro lado el diseño es centrado en casos de uso y enfocado al negocio, con soporte para los Diagramas de Procesos de Negocios (BPD) y Diagramas de Flujos de Datos (DFD), lo cual genera un software de mayor calidad.

Usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación y esta capacitado para la ingeniería directa e inversa en Java, C++, PHP, además de la capacidad de generación de código en estos lenguajes.

Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Específicamente presenta dos tipos de diagramas de modelación de bases de datos: entidad-relación (ERD) y mapeo objeto relacional (ORM). Los diagramas ERD modelan la base de datos a nivel físico y los ORM muestran la relación entre las clases (orientado a objeto) y la entidad (de la base de datos). Esta característica permite generar además del script de la base de datos, el código de las clases persistente (o clases de entidad) en el lenguaje escogido.

Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al momento, creando de forma simple toda la documentación.

Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama. A continuación en los subtópicos que suceden se relacionan el resto de las herramientas y tecnologías a utilizar agrupadas por los siguientes criterios: programación del lado del cliente, del lado del servidor y las relacionadas con base de datos.

2.2.3 METODOLOGÍA DE DESARROLLO

Proceso de Desarrollo y Gestión de Proyectos de Software V1.0.

El modelo de desarrollo de software propuesto describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Se logra con la combinación entre los modelos basado en Componentes, el Iterativo y el Incremental. Se emplearán las técnicas de prototipado, si son requeridas, para los requerimientos del usuario de los que no existe una visión clara por parte de estos, con el objetivo de desarrollar una definición mejorada de los requisitos del usuario para el sistema.

Desarrollo iterativo e incremental: Es un enfoque en el que el ciclo de vida está compuesto por iteraciones, estas son pequeños procesos compuestos de varias actividades cuyo objetivo es entregar una parte del sistema parcialmente completo, probado, integrado y estable. Todo el software es integrado en cada entrega de cada iteración hasta obtener el producto de software completo en la última iteración. En cada iteración se obtiene como resultado un incremento.

Desarrollo basado en componentes: Nos lleva a alcanzar un mayor nivel de reutilización de software, aún en contextos distintos a aquellos para los que fue diseñado. Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema. Dado que un componente puede ser construido y luego mejorado continuamente, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.

2.2.4 PROGRAMACIÓN DEL LADO DEL CLIENTE

En la capa de presentación, del lado del cliente donde se implementan todas las interfaces y componentes visuales que posibilitan brindarle información al usuario, se utilizan las siguientes tecnologías y herramientas libres:

- **Mozilla Firefox** 3.0 ó superior (Recomendado Firefox 3.6, con soporte para los lenguajes de programación utilizados en esta capa: HTML 5, CSS 3, JavaScript 1.8.2 y JSON estándar). Mozilla Firefox es un navegador web libre que sirve para visualizar páginas web, es multiplataforma y está disponible en varias versiones de Microsoft Windows, Mac OS X, GNU/Linux y algunos sistemas basados en Unix. Su código fuente es software libre y está publicado bajo una triple licencia GPL/LGPL/MPL. Usa el motor de renderizado Gecko, que implementa algunos estándares web actuales además de otras funciones, algunas de las cuales están destinadas a anticipar

probables adiciones a los estándares web. Incluye navegación por pestañas, corrector ortográfico, búsqueda progresiva, marcadores dinámicos, un administrador de descargas y un sistema de búsqueda integrado que utiliza el motor de búsqueda que desee el usuario. Además se pueden añadir funciones a través de complementos desarrollados por terceros. Es muy rápido, seguro, estable y de fácil personalización. Se le integran muchas extensiones útiles en tiempo de programación.

- **HTML** (HyperText Markup Language) es un lenguaje de marcación de elementos para la creación de documentos hipertexto. Permite combinarse con otros lenguajes para definir el formato con el que se tienen que presentar las webs, como CSS y JavaScript. Este lenguaje se encarga de indicar a los navegadores cómo deben mostrar el contenido de una página web.
- **CSS** (Cascading Style Sheets), Hojas de Estilo en Cascadas, es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Estas hojas permiten el control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo y los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario.
- **JavaScript** es un lenguaje de programación del lado del cliente, interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Es un lenguaje orientado a objetos, ya que dispone de Herencia, aunque la realiza siguiendo el paradigma de programación basada en prototipos, pues las nuevas clases se generan clonando las clases base y extendiendo su funcionalidades un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.
- **ExtJS** 2.2 ó 3.1 (Recomendado ExtJS 2.2 para la creación de componentes a nivel de interfaz. ExtJS es un framework o librería construido con JavaScript que concentra su

potencia en la rica colección de componentes para el diseño de GUI's (Grafic User Interfaces) complejas y dinámicas del lado del cliente, haciendo uso extensivo de Ajax. Entre los componentes que este framework ofrece encontramos cuadros de diálogo, menús, tablas editables, capas, paneles, pestañas y todo lo necesario para construir atractivos desarrollos al estilo de Web 2.0. La misma da soporte para comunicar datos de forma asíncrona con el servidor y manejarlos aunque sean de distinta índole de una manera simple. Actualmente Ext.JS es considerado un framework independiente; ya que a principios del 2007 se creó una compañía para comercializar y dar soporte al mismo, dicha compañía proporciona los servicios de consultoría necesarios para ayudar a los clientes en el aprovechamiento máximo de las ventajas de Ext.JS.

Amparado bajo GPLv3, Ext. 2.0 es un framework completo y extremadamente avanzado, que contiene casi todo lo que te puedas imaginar y nace como solución a tareas comunes -pero complejas. Este framework te hará recordar la programación de entornos visuales de PC, ya que es completamente basado en OO. Cada objeto contiene lo típico: propiedades, métodos, eventos...etc. Ext.JS basa toda su funcionalidad en JavaScript a través de librerías ya muy conocidas: YUI, jQuery y Prototype/Script.aculo.us y un núcleo interno poderoso. Así, en tiempo de ejecución carga y crea todos los objetos html a través del uso intenso de DOM.

Ventajas:

- ✓ La orientación a objetos intensa te hará modular todos tus scripts.
- ✓ El diseño está completamente separado de la funcionalidad.
- ✓ Funciones comunes como validación, combobox's editables, ventanas desplazables (con minimizar y maximizar), grillas editables, son muy fáciles de implementar.
- ✓ Buena y amplia documentación, así como también su comunidad.

2.2.5 PROGRAMACION DEL LADO DEL SERVIDOR

Para la programación del lado del servidor donde se integra la capa de presentación con las capas de negocio se emplean las siguientes tecnologías y herramientas.

- **Zend Framework** 1.9.7 se trata de un framework para desarrollo de aplicaciones Web y servicios Web con PHP, te brinda soluciones para construir sitios web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5.

Es una implementación que usa código 100% orientado a objetos, donde cada uno de sus componentes está construido con una baja dependencia entre ellos. Esta arquitectura, débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

Este framework ofrece un gran rendimiento y una robusta implementación del modelo vista controlador, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la presentación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Cualesquiera que sean las necesidades de una solicitud, usted tiene todas las posibilidades de encontrar un componente de de Zend Framework que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida. [\[35\]](#)

- **ZendExt** 1.5.4 Beta (Extensión de Zend Framework para satisfacer las características propias de la arquitectura de sistema de las aplicaciones de la UCID, extendiendo en el MVC Zend_Controller a través de ZendExt_Controller).

ZendExt 1.5.4 Beta (implementando ZendExt_Model para la implementación del negocio de las aplicaciones).[\[36\]](#)

En la capa de servicio se utiliza ZendExt 1.5.4 Beta (implementando ZendExt_loC para la integración de sistemas, módulos y componentes de forma no distribuida).

- **AJAX** (Asynchronous JavaScript and XML) El término AJAX creado en el año 2005 no es una tecnología en si misma, sino la unión de varias tecnologías que trabajan conjuntamente; el concepto general es: cargar y renderizar una página, luego mantenerse en esa página mientras scripts y rutinas van al servidor buscando, en background, los datos que son usados para actualizar la página solo re-renderizando la página y mostrando u ocultando porciones de la misma. Es decir en un lenguaje más asequible sería: técnica de desarrollo web para crear aplicaciones interactivas que se ejecutan del lado del cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. La mayoría de los navegadores actuales permiten la utilización de AJAX así que no constituye un problema mayor la compatibilidad con éstos, a excepción de unos pocos utilizados en menor medida.

Tecnologías que integra AJAX:

- ✓ XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- ✓ Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario como pueden ser JavaScript o JScript para mostrar e interactuar dinámicamente con la información presentada.

✓ El objeto `XMLHttpRequest` para intercambiar datos asincrónicamente con el servidor web.

✓ XML es el formato usado comúnmente para la transferencia de vuelta al servidor.

En resumen una aplicación web que use AJAX se diferencia de las clásicas en que ésta introduce un intermediario, un motor AJAX entre el usuario y el servidor. En vez de cargar una página Web, al inicio de la sesión, el navegador carga al motor AJAX, que es el responsable de renderizar la interfaz que el usuario ve y de comunicarse con el servidor en nombre del usuario. El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo.

➤ **Apache:** En lo que se refiere a las tecnologías por parte del servidor sobresale Apache programa que administra servidores para alojar sitios Web. Según el fabricante, alrededor del 50 por ciento de los sitios de Internet está almacenado en servidores Apache.

Es estable, seguro y eficiente, software libre y de código abierto para plataformas Unix, Windows y otras, presenta una arquitectura muy modular.

Presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Apache es un servidor de red para el protocolo HTTP, elegido para poder funcionar como un proceso independiente, sin que eso solicite el apoyo de otras aplicaciones o directamente del usuario.

Trabaja con Perl, PHP y otros lenguajes de script. Apache utiliza Perl tanto con soporte CGI como con soporte modo Perl. También trabaja con Java y páginas JSP, teniendo todo el soporte que se necesita para tener páginas dinámicas. Te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. Tiene una alta configurabilidad en la creación y gestión de logs.

Permitiendo la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

Apache es llamado el servidor Web por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

➤ **PHP** es un lenguaje interpretado del lado del servidor 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. Es un lenguaje multiplataforma con capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, tiene capacidad de expandir su potencial utilizando la enorme cantidad de módulos, llamados extensiones, es libre, por lo que se presenta como una alternativa de fácil acceso para todos, además de que permite la utilización de las técnicas de Programación Orientada a Objetos.

Las cuatro grandes características de PHP: Velocidad, estabilidad, seguridad y simplicidad.

✓ **Velocidad:** No solo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes Unix, cuando se configura como módulo de Apache, esta listo para ser utilizado.

✓ **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de bugs, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho mas difícil para lo bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.

✓ **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, estos pueden ser configurados desde el archivo .ini.

✓ **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible.

➤ **Zend Studio** o Zend Development Environment (IDE) es un completo entorno integrado de desarrollo para el lenguaje de programación PHP. Está escrito en Java, y es multiplataforma, disponible para Microsoft Windows, Mac OS X y GNU/Linux. Soporta PHP4 y PHP5, presenta resaltado de sintaxis, plegado de código, inserción automática de paréntesis y corchetes de cierre, detección de errores de sintaxis en tiempo real entre otras muchas funcionalidades que agilizan el trabajo del programador. Incluye además funciones de errores de depuración permitiendo dicha

acción en servidores remotos. Soporte para la gestión de grandes proyectos, para el control de versiones y para la navegación en bases de datos y ejecución de consultas SQL.

2.2.6 BASE DE DATOS

- **Doctrine** como mapeador relacional de objetos (ORM) para el PHP 5.2.3+ que se encuentra en la parte superior de una capa de abstracción de base de datos poderosa (DBAL). Una de sus características claves es la opción para escribir las consultas de base de datos en un objeto propietario orientado SQL, llamado Doctrine Query Language(DQL), inspirado por Hibernate HQL. Esto proporciona desarrolladores con una alternativa poderosa a SQL que mantiene flexibilidad sin requerir duplicación de código innecesaria. Otra característica que posee es la posibilidad de exportar una base de datos existente a sus clases correspondientes y viceversa.
- **PostgreSQL** como Sistema Gestor de base de datos (SGBD) que a través de un conjunto de programas permite crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:
 - ✓ Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
 - ✓ Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD
 - ✓ Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Dentro de los gestores de bases de datos existentes, se nombra como uno de los más distintivos a PostgreSQL, un motor de base de datos que es servidor de base de datos relacional libre. El mismo está considerado como la base de datos de código abierto más avanzada del mundo, proporcionando un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. Está diseñado para soportar volúmenes masivos de datos, sin que ello afecte en lo absoluto en su rendimiento. Ejemplo de ello es que puede soportar tuplas de hasta 1600 campos y retornar a su vez millones de estas tuplas en apenas unos segundos.

Ofrece una fortaleza adicional sustancial al incorporar cuatro conceptos adicionales básicos: Clases, Herencia, Tipos, Funciones. Cuenta además con características que aportan potencia y flexibilidad adicional: Restricciones, Disparadores, Reglas, Integridad transaccional.

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario. Soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. Posee soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Corre en la casi totalidad de los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows, etc. La documentación está muy bien organizada, pública y libre, con comentarios de los propios usuarios. Las comunidades muy activas, con varias de ellas en español. Soporta el protocolo de comunicación encriptado por SSL. Posee utilidades para limpieza de la base de datos (Vacuum) y para el análisis y optimización de Query's.

2.2.7 APLICACIONES, MÓDULOS Y COMPONENTES DE SAUXE 1.5.4 BETA

- ✓ Acaxia – 2.5.2 Beta
- ✓ SIDEC – 2.0.5 Beta
- ✓ Portal – 1.5.3 Beta
- ✓ Trazas – 1.0.1 RC

Tabla 3: Herramientas y tecnologías de desarrollo.

No.	Nombre y versión	Sitio web	Incluida	Distribuida
	Postgres v.8.3.5	http://www.postgresql.org/		Sí
	PHP v.5.2.4	http://www.php.net/	Sí	
	Apache HTTP Server v.2.2	http://www.apache.org/		Sí
	PHPUnit v.3.2.20	http://www.phpunit.de/	Sí	
	PHPDocumentator v.1.4.0	http://www.phpdoc.org/	Sí	
	Zend Framework v.1.7.3	http://framework.zend.com/	Sí	
	ZendExt Framework v.1.5	Desarrollo	Sí	
	ExtJS v.2.2	http://www.extjs.com/	Sí	
	Zend Studio for Eclipse v.6.0	http://www.zend.com/en/		Sí
	Doctrine ORM v.1.1.3	http://www.doctrine-project.org/	Sí	

	Visual Paradigm v.6.3	http://www.visual-paradigm.com/		Sí
	SVN v.1.4.6	http://subversion.tigris.org/		Sí
	Mozilla Firefox v. 2.0.17	http://www.mozilla.org/		Sí
	Estructura y Composición v.1.0	Desarrollo	Sí	
	Traza v.1.0	Desarrollo	Sí	
	Portal v.1.0	Desarrollo	Sí	

2.3 MODELADO DE LA SOLUCIÓN

2.3.1 ANALISIS DEL ENTORNO DEL PROBLEMA

Durante el desarrollo de las acciones combativas se dan muchísimas situaciones ante las cuáles el jefe debe decidir qué hacer en muy corto tiempo y de forma eficiente, de manera tal que garantice la protección de los recursos humanos y materiales que tiene subordinados para poder continuar el combate. En ese momento, el jefe como ser racional se ve afectado por el stress y otros factores humanos, que en muchas ocasiones impiden que tome la mejor decisión, de ahí que necesite apoyarse en un sistema que liberado de todos esos riesgos, procese la información necesaria para decidir en cuestiones de segundos y pueda ser utilizada por todos los jefes desde el nivel de Ejército hasta nivel de Batallón.

2.3.2 MODELO CONCEPTUAL

A través del siguiente modelo conceptual se representan visualmente los conceptos u objetos del mundo real significativos para el problema tratado. Se enfoca en clases conceptuales del dominio del problema objeto de estudio. Las principales clases conceptuales a las que se hace referencia son: base de casos, caso, atributo y valor.

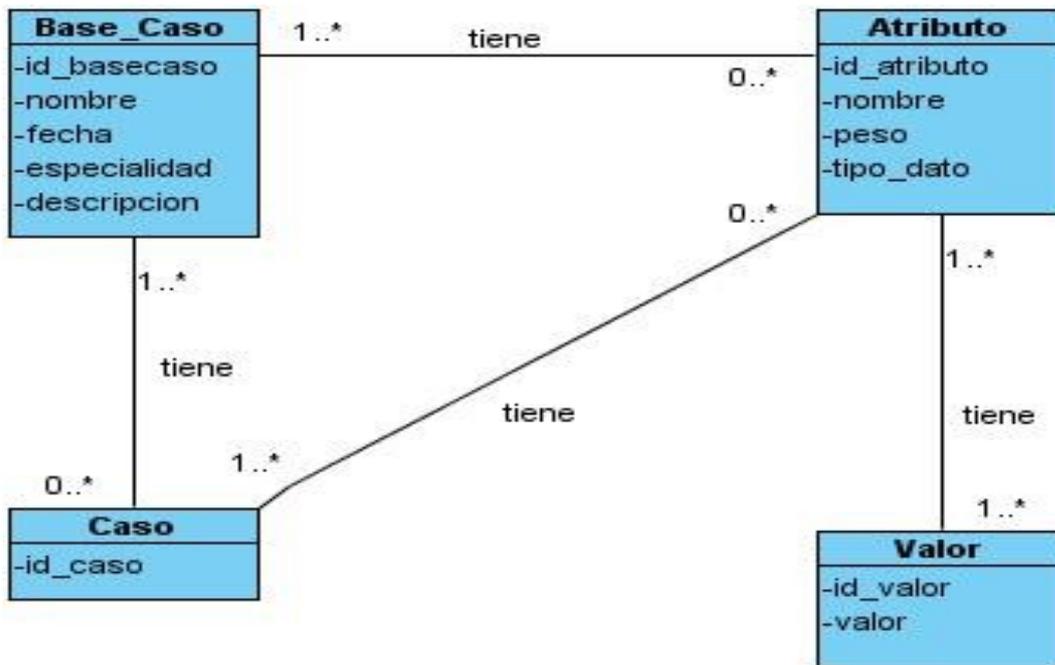


Fig.10: Modelo Conceptual

En este diagrama se visualiza las entidades que intervienen durante el proceso de toma de decisiones. Estas entidades pueden resultar posibles tablas de la base de datos.

2.3.3 REQUERIMIENTOS FUNCIONALES

RF1: Gestionar Base de Casos.

RF1.1-Adicionar Base de Casos.

RF1.2-Modificar Base de Casos.

RF1.3-Eliminar Base de Casos.

RF1.4-Buscar Base de Casos.

A través del requisito funcional Gestionar Base de Casos el sistema brinda la posibilidad de adicionar una nueva base, a la cual se le define el nombre, una especialidad o área que comprende, un conjunto de atributos asociados (aspectos que va a tener en cuenta el sistema para tomar la decisión), una breve descripción y la fecha en que fue elaborada; de modificar una base de casos ya existente o eliminarla en caso que sea necesario. También va a permitir buscar una base de casos dentro de una lista de bases, teniendo en cuenta diversos criterios.

RF2: Gestionar Casos.

RF2.1-Adicionar Caso.

RF2.2-Modificar Caso.

RF2.3-Eliminar Caso.

El requisito funcional Gestionar Casos facilita a los expertos introducir el conocimiento y las experiencias a la base a través de la adición de nuevos casos, modificar y/o eliminar casos ya registrados con anterioridad. A través de la agregación de nuevos casos el sistema es capaz de aprender y por ende mostrar un comportamiento más inteligente.

El objetivo principal de este requisito es conformar el repositorio o base de conocimientos para tener indicadores o un patrón de comportamiento, que permita rápidamente visualizar tendencias, posiciones y ubicaciones con respecto a una planeación original teniendo en cuenta diferentes criterios (atributos de la base de casos) con el fin de tener los elementos para tomar una decisión y perfeccionar los procesos para el control de riesgos.

RF3: Gestionar Atributos de la Base de Casos.

RF3.1-Adicionar Atributo.

RF3.2-Modificar Atributo.

RF3.3-Eliminar Atributo.

El requisito funcional Gestionar Atributos permite definir cuáles serán las propiedades o atributos que manejará la base, modificar atributos ya introducidos y/o suprimirlos.

RF4: Gestionar Dominio.

RF4.1-Adicionar Valor Dominio.

RF4.2-Modificar Valor Dominio.

RF4.3-Eliminar Valor Dominio.

El requisito funcional Gestionar Dominio proporciona al usuario definir el conjunto de valores admisibles que puede tomar cada atributo de la base de casos, así como modificarle un valor determinado o eliminárselo.

RF5: Inferir Decisiones.

Este requisito es el que permite consultar y procesar toda la información contenida en la base de casos. Para ello considera casos anteriores y entonces llega a una decisión en dependencia de la comparación entre la situación actual y los casos viejos. Dado un nuevo caso para clasificar, el sistema verifica primero si existe un caso de entrenamiento idéntico. Si no encuentra ningún caso idéntico, el razonador buscará casos de entrenamiento que tienen componentes similares a aquellas del nuevo caso y brindará la solución del que más se asemeja. Es decir, se busca el caso que tenga

más propiedades con valores coincidentes con el caso que se analiza, devolviendo como respuesta el valor que tiene la propiedad buscada en el caso encontrado.

RF6: Buscar similitud entre casos.

El requisito funcional Buscar similitud ofrece al usuario la posibilidad de visualizar cuáles son los casos más parecidos a la situación actual de manera descendente y el porcentaje de similitud.

2.3.4 FORMULAS Y ECUACIONES

El ciclo del Razonamiento Basado en Casos está formado por los siguientes pasos:

1. Recuperación.
2. Reutilización.
3. Revisión.
4. Retención o Almacenamiento.

La **recuperación de casos** es la selección, en la base de conocimiento, de aquellos casos cuya descripción se ajusta más a la información presentada en el nuevo caso.

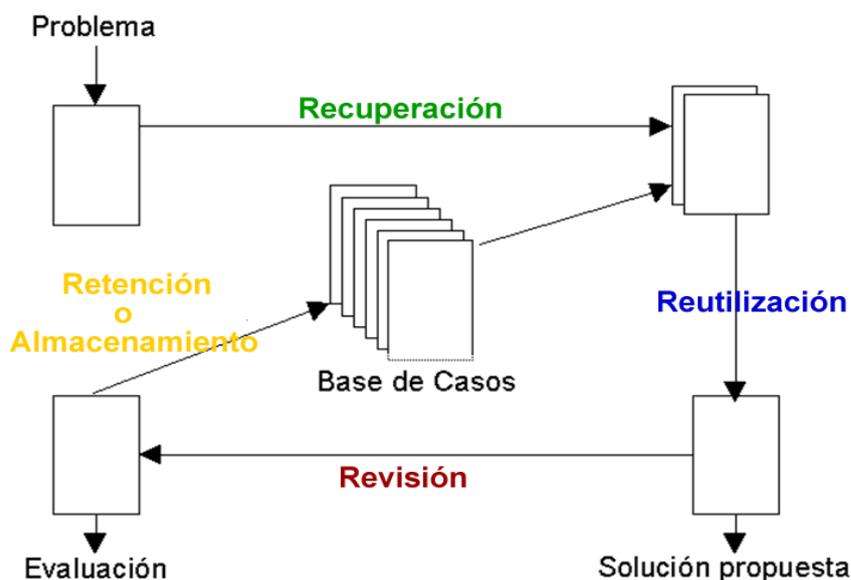


Fig. 11: Diagrama conceptual del proceso de razonamiento basado en casos.

Para ellos las técnicas más comúnmente utilizadas son: vecinos más cercanos y recuperación inductiva.

La técnica de “Vecino más cercano” consiste en determinar la distancia que existe entre las características del nuevo caso y las de los casos ya existentes, localizando el caso que esté más cerca.

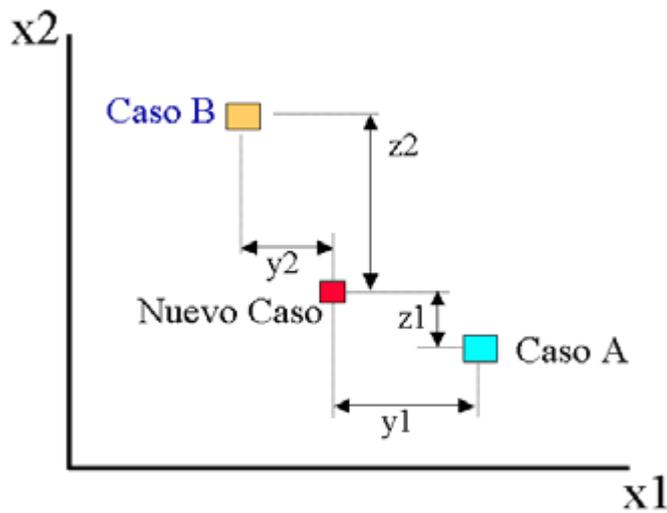


Fig. 12: Diagrama que muestra la distancia entre el nuevo caso y los casos A y B. X1 y x2 son las características que definen a los casos.

Normalmente para destacar una característica, se pondera su distancia, con un peso w_i , para hacer que se acerque o parezca más a uno en particular, como se muestra en la ecuación siguiente:

$$\text{Similitud}(T, S) = \sum_{i=1}^n f(T_i, S_i) \times w_i$$

$$\text{semejanza}(\text{Caso}, \text{CasoR}) = \frac{\sum_{i=1}^n w_i \cdot \text{sim}(f_i^L, f_i^R)}{\sum_{i=1}^n w_i}$$

donde f es una función de similitud, T es nuevo caso, S es un caso previo y n es el número de atributos del caso.

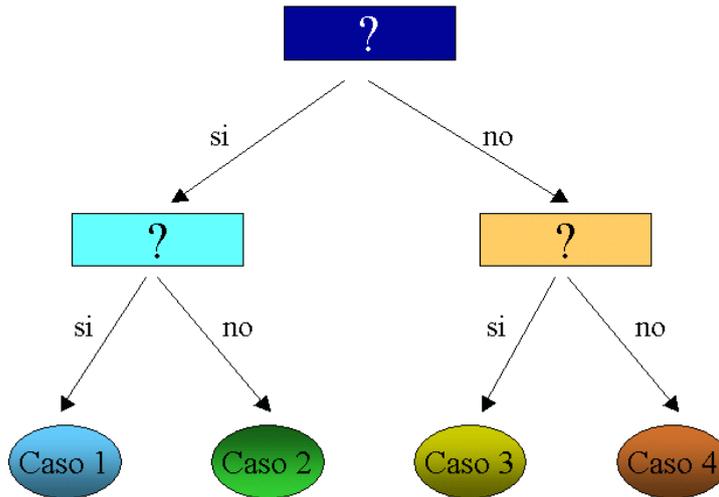


Fig.13: *Árbol de decisiones para la clasificación de casos.*

La técnica de “Recuperación Inductiva” requiere de la preparación previa de un árbol de decisiones, a partir del cual, se encuentre el caso más parecido a la información dada. El inconveniente es que si no se tienen todas las respuestas, no se puede localizar un único caso parecido..

La **reutilización** consiste en adecuar la solución del caso más parecido a las condiciones del nuevo caso. Esto es necesario, dado que normalmente los fenómenos o síntomas que se presentan en un diagnóstico, no son idénticos a los ocurridos en los casos anteriores.

La etapa de **revisión**, consiste en la validación de la solución propuesta. Esta validación se realiza contrastando diferentes soluciones o simulando la solución para estimar qué tan acertada es. Por lo tanto, esta etapa está altamente influenciada por el grado de conocimiento que tiene el experto sobre el fenómeno ocurrido, y es él el que juzga la efectividad de la solución propuesta con base en su experiencia. También durante esta etapa se aplica la solución propuesta y se analiza el resultado de su aplicación. Si los resultados son los esperados se confirma la solución, pero si existen diferencias, se debe averiguar por qué ocurrieron tales diferencias y cómo pueden evitarse. Esta información debe servir para mejorar la definición del caso.

Finalmente, **la retención o almacenamiento** consiste en registrar, en la base de conocimiento, la información derivada del nuevo caso, ya sea como un caso nuevo o un caso mejorado.

2.3.5 REQUERIMIENTOS NO FUNCIONALES

Los requisitos no funcionales son condiciones que debe cumplir un sistema para satisfacer un contrato o una especificación. Están regidos por las necesidades del usuario para poder resolver un problema o conseguir un beneficio determinado. Se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida, y la representación de datos que se utilizan en las interfaces del sistema.

Estos requisitos son de gran significación en la aceptación del software, debido a que representan las ventajas más visibles al usuario y repercuten en el óptimo funcionamiento y mantenimiento del sistema.

Apariencia o interfaz externa

La interfaz externa (vista que se le presentará al usuario que utilice el sistema), como intermediaria entre el software y el usuario debe ofrecer facilidades de entendimiento y sencillez al realizar las operaciones que en ella se presentan. La interfaz debe ser lo más sencilla posible, para que pueda ser manejada por cualquier tipo de usuario. Debe presentar una buena combinación de colores .Debe mantener la interfaz principal del sistema implementado anteriormente; logrando que el cliente se sienta identificado con la aplicación.

Usabilidad

El sistema debe poder ser usado por cualquier persona que tenga conocimientos básicos de informática y del proceso de realización del combate. El software contará con una ayuda, que se mantendrá visible para apoyar el empleo del sistema.

Seguridad

Autenticación que es la contraseña de acceso. Autorización que no es más que darle atribuciones a los usuarios respecto a sus funciones de trabajo. Implementación de auditoria para registrar la confirmación de cada operación efectuada por el usuario.

Rendimiento

Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.

Soporte

Para el servidor de aplicaciones: Se requiere que esté instalado un intérprete de ficheros PHP rápido y con las últimas actualizaciones del lenguaje. Para el servidor de base de datos: Se requiere que esté instalado un gestor de base de datos que soporte grandes volúmenes de datos, maneje la concurrencia y transacciones. Para el cliente: Se requiere que esté instalado un navegador que interprete JavaScript.

Portabilidad

El sistema debe ser multiplataforma haciéndose énfasis en la plataforma Linux.

Legales

El sistema se basará en el manual de normas y principios establecidos por el MINFAR.

Software

Para el cliente:

1. Navegador Mozilla Firefox.
2. Sistema operativo Windows XP o superior o Linux.

Hardware

Requerimientos máximos para el servidor:

1. Computador Pentium4.
2. 512 MB RAM o superior.

Requerimientos mínimos para la conexión del cliente:

1. Computador Pentium4 o superior.
2. 256 MB RAM o superior.

Aplicación de Estándares

Se aplicarán los estándares de ingeniería, codificación, diseño para la base de datos y mecanismos de diseño definidos por UCID.

Políticos Culturales

El sistema podrá ser usado por diversas entidades durante el proceso de toma de decisiones, para ella cada entidad deberá configurarlo teniendo en cuenta sus características. El software debe ser en idioma español.

Requerimientos de ayuda y documentación

Se propone que el sistema cuente con una ayuda general en la página principal, que guiará al usuario de cómo trabajar en el sistema, también estará disponible en cada una de las interfaces, de esta forma los usuarios tendrán conocimiento de las

funcionalidades del mismo y hacer un mejor uso de estas.

Restricciones de diseño

El producto de software final debe diseñarse sobre una arquitectura modelo-vista-controlador, con un estilo en capas. Emplear los estándares establecidos (diseño de interfaces, base de datos y codificación). Emplear como lenguaje del lado del servidor, PHP y del lado del cliente Javascript.

2.4 DISEÑO DE LA SOLUCIÓN

El diseño del software se encuentra en el núcleo técnico de la ingeniería del software y se aplica independientemente del modelo de diseño de software que se utilice. Una vez que se analizan y especifican los requisitos del software, el diseño del software es la primera de las tres actividades técnicas (diseño, generación de código y pruebas) que se requieren para construir y verificar el software. Cada actividad transforma la información de manera que dé lugar por último a un software validado.

2.4.1 DISEÑO ARQUITECTONICO

El propósito de realizar el diseño arquitectónico del software es estructurar los componentes del sistema, sus interrelaciones, y los principios y reglas que gobiernan su implementación y evolución en el tiempo. Aporta además una visión abstracta de alto nivel, postergando el detalle de cada uno de los módulos definidos a pasos posteriores del diseño. Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema. Para el desarrollo de las funcionalidades se utiliza el marco de trabajo definido por el centro UCID y estandarizado para todos los proyectos.

2.4.1.1 PATRÓN ARQUITECTÓNICO MODELO VISTA CONTROLADOR (MVC)

Para la implementación de la solución se propone utilizar el estilo en capas (capa de presentación, capa de negocio y capa de acceso a datos) y el patrón MVC el cual está formado por tres niveles:

- El modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio. Integra las clases que contienen las consultas a la Base de Datos (Modelo de Negocio), así como la definición de las tablas, sus relaciones y atributos.

- La vista transforma el modelo en una página web que permite al usuario interactuar con ella. Determina la interfaz que se muestra finalmente al cliente para su intercambio con la aplicación.
- El controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista. Gestiona todas las peticiones del usuario y se encarga de darles respuesta.

El patrón de arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, y otros.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación.

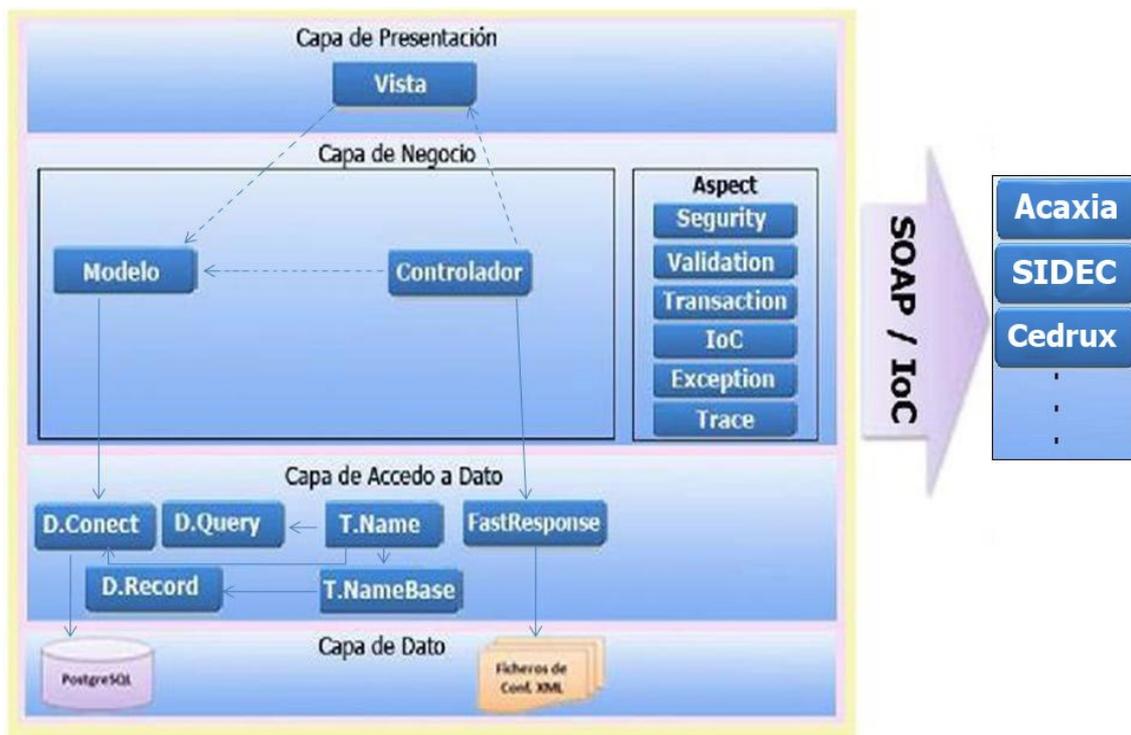


Fig.14: Vista a nivel macro de la arquitectura.

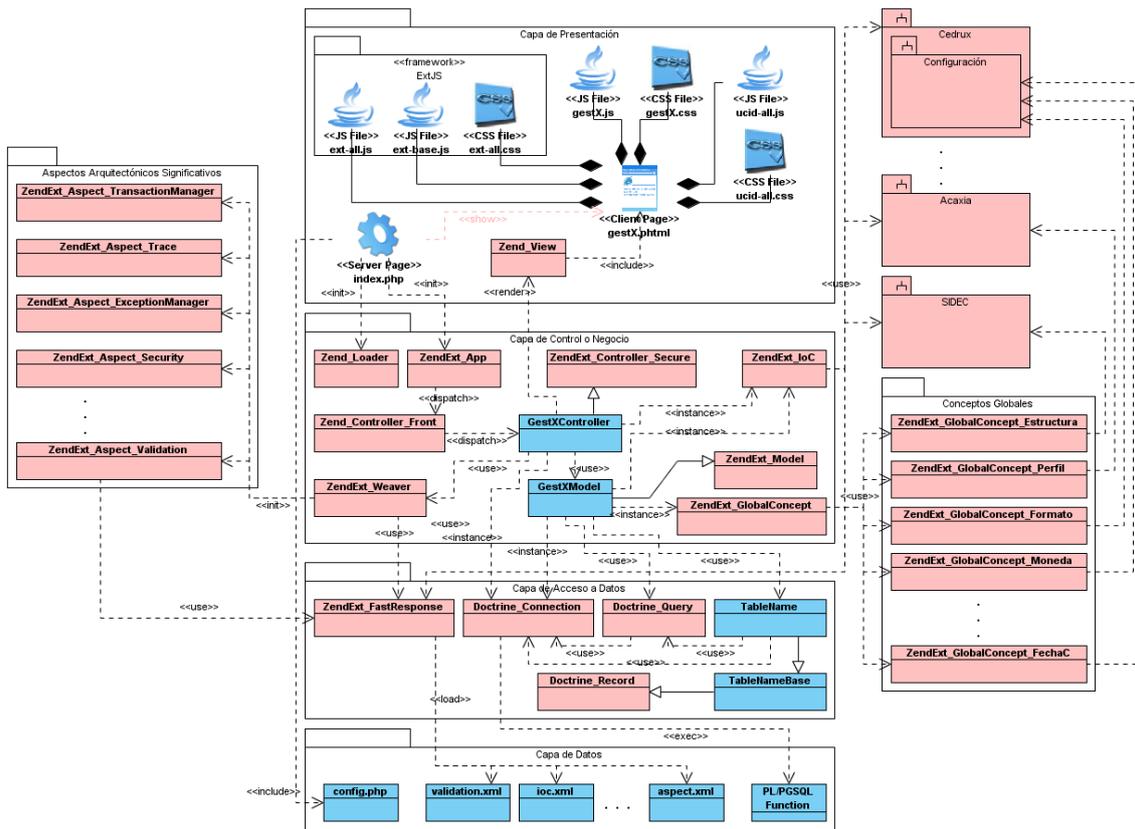


Fig.15: Modelo Arquitectónico.

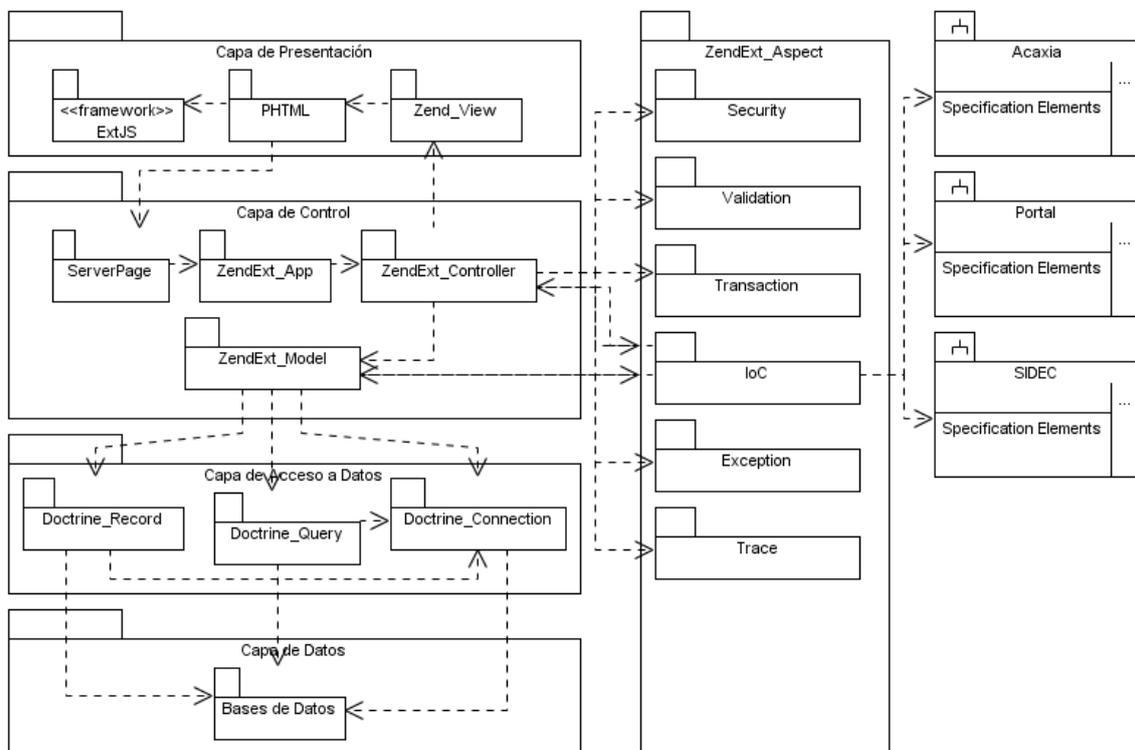


Fig. 16: Estilo Arquitectónico.

2.4.1.2 PATRONES DE DISEÑO

2.4.1.2.1 Patrones GoF

Singleton (Instancia única): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Factory Method: Define una interfaz para crear un objeto, pero deja que sean las subclases quienes decidan qué clase instanciar. Permite que una clase delegue en sus subclases la creación de objetos.

Table Data Gateway (TDG): Un objeto que actúa como Gateway (Puerta de enlace) a la base de datos. Una instancia que maneja todas las filas en una tabla. Table Data Gateway contiene todo el SQL para acceder a una sola tabla o vista. Otro código invoca sus métodos para interactuar con las base de datos.

Row Data Gateway (RDG): Permite disponer de un objeto que representa directamente una fila de la base de datos. Todo el acceso a la fila queda oculto tras la interfaz de la clase. En principio, no representa mayor lógica de negocio.

2.4.1.2.2 Patrones GRASP

Experto: Este se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Creador: Este patrón se encarga de asignarle a la clase B la responsabilidad de crear una instancia de clase A. B es un creador de los objetos A.

Alta Cohesión: Este patrón se encarga de asignar una responsabilidad de modo que la cohesión siga siendo alta.

Bajo Acoplamiento: Este patrón se encarga de asignar una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible.

Front Controller: implica que todas las solicitudes son dirigidas a un único script PHP que se encarga de instanciar al controlador frontal y redirigir las llamadas.

Decorator: encargada de asignarle responsabilidades a objetos de manera dinámica y configurarlos con nuevos atributos.

2.4.1.3 DIAGRAMA DE COMPONENTES

Los diagramas de componentes permiten describir los elementos físicos que integran el sistema y las relaciones que existen entre ellos. Muestran además las opciones de realización incluyendo código fuente, binario y ejecutable.

Los componentes representan todos los tipos de elementos software que entran en la producción de aplicaciones informáticas. Pueden ser simples archivos, paquetes, y/o bibliotecas cargadas dinámicamente.

A continuación se exponen los Diagramas de Componentes asociados a varios de los subsistemas de implementación identificados. Siguiendo la arquitectura en capas, la estructuración en subsistemas de implementación es la siguiente:

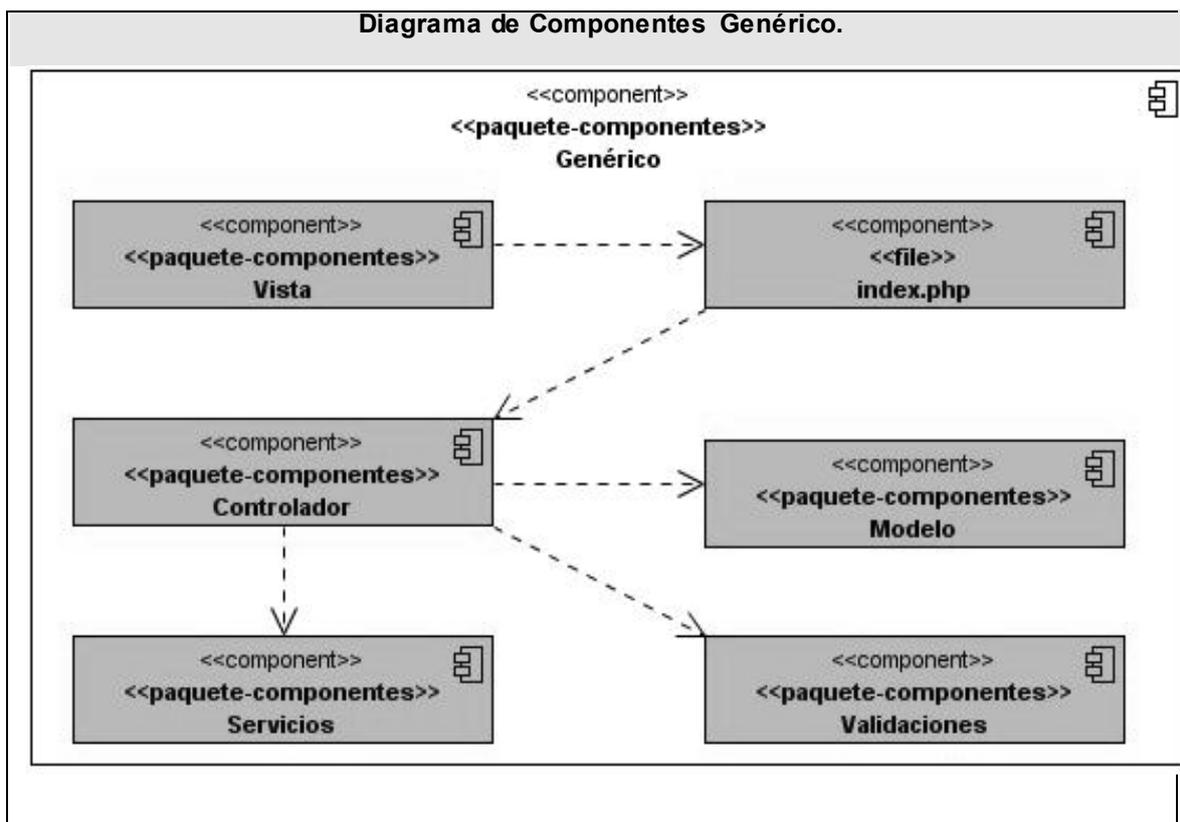


Fig.17: Diagrama de paquetes de componentes por subsistemas genérico.

En este diagrama de componentes genérico se representan las relaciones entre las diferentes capas de la arquitectura.

A continuación se representa el Diagrama de componentes del módulo de gestión del conocimiento (inteligencia artificial) para la toma de decisiones, en el cual se especifican las relaciones de ese subsistema con el resto de los subsistemas existente, detallando los servicios que él consume tanto externos como internos.

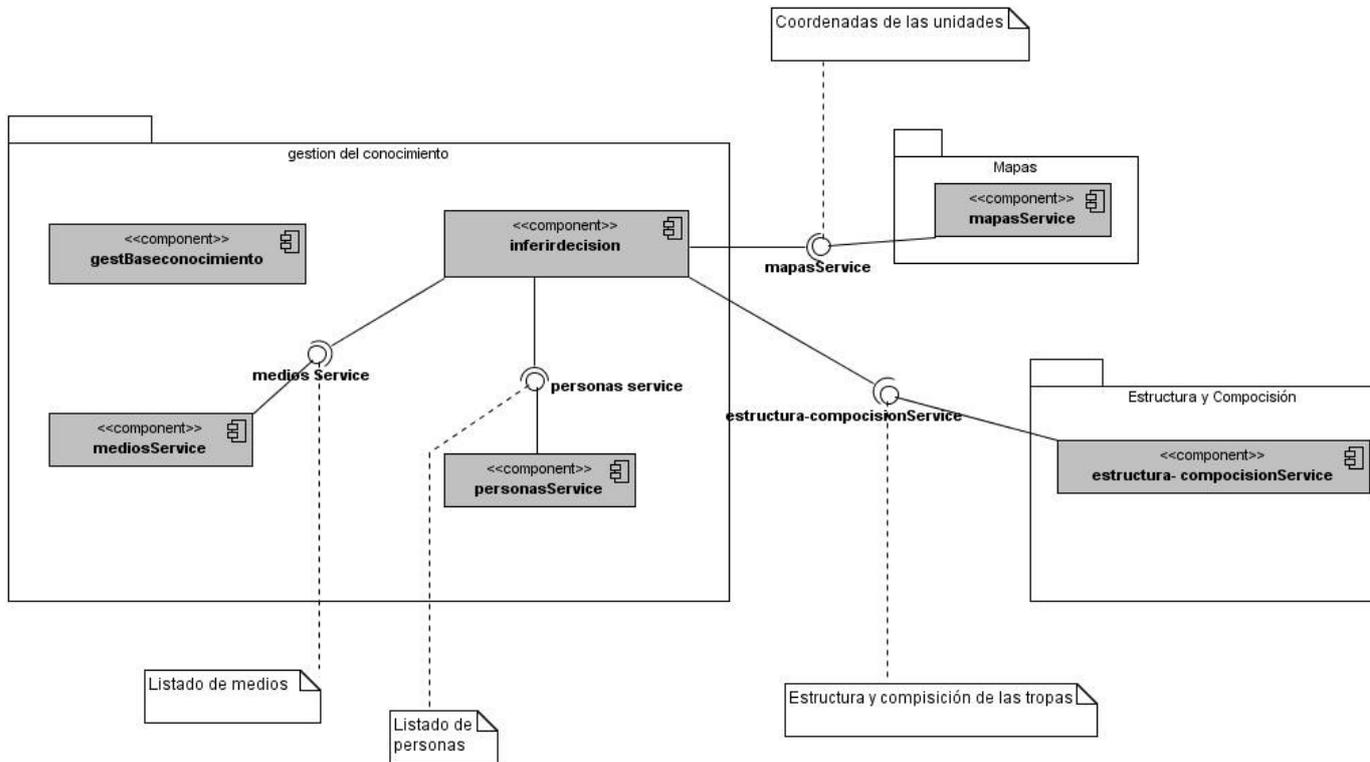
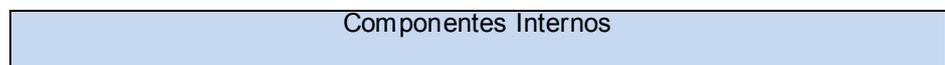


Fig.18: Diagrama de paquetes de componentes del subsistema de gestión del conocimiento.

2.4.1.4 MATRIZ DE INTEGRACION DE COMPONENTES

La matriz de integración de componentes contiene todos los componentes definidos en el subsistema, de forma matricial, y en las intercepciones se especificarán cada uno de los servicios que consume el componente. Existen dos matrices de integración de componentes: una matriz interna donde se especifica la integración entre los componentes internos del sistema, y una externa donde se define la integración con aquellos componentes que no pertenecen al sistema y de los cuales consumimos determinados servicios.



Componentes Internos	Gestionar Base de Conocimiento	Inferir decisión	Medios	Personas
Gestionar la base de conocimientos	Crear Base de conocimientos. Modificar Base de Conocimientos. Eliminar Base de Conocimientos	No	No	No
Inferir decisión	No	Recuperar Casos de la Base de Conocimientos. Inferir decisión.	Listado de medios	Listado de Personas

Tabla 4: Matriz de integración de componentes internos.

3.1

Componentes Internos	Componentes Externos	
	Estructura y Composición	Mapas
Gestionar la base de conocimientos	No	No
Inferir decisión	Listado de la estructura y la composición de las tropas.	Ubicación y Coordenadas de las unidades.

Tabla 5: Matriz de integración de componentes externos.

2.4.2 DISEÑO DE LOS DATOS

2.4.2.1 DIAGRAMA DE CLASES DE LA BASE DE DATOS

A través del modelo de datos se definen los conceptos que se manejan en el sistema y que sirven para describir la estructura de la base de datos diseñada. Es decir, en dicho modelo se representan los datos, sus atributos y tipos, sus relaciones y las restricciones que deben cumplirse sobre ellos.

A continuación se presenta el modelo de datos para el módulo gestión del conocimiento donde se definen las tablas: `dat_basecaso` que hace referencia a la base de conocimientos, `dat_caso` que representa cada caso de la base de casos, `dat_atributo` que hace alusión a los atributos que tiene cada base de casos y `nom_valordominio` que es la tabla que almacena todos los valores posibles que pueden tomar los atributos. Estas son las principales tablas del esquema elaborado, además de otras que surgen como resultado de las relaciones entre las primeras.

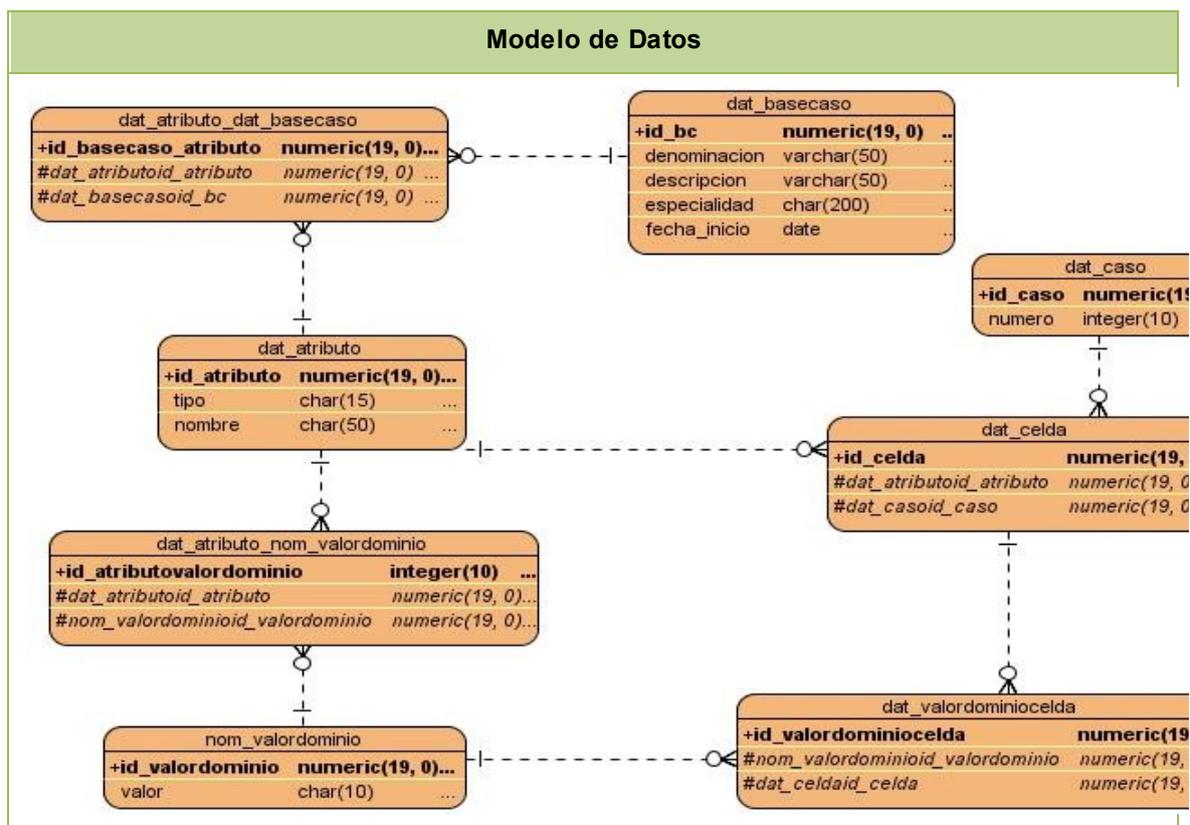


Fig.19: Modelo de Datos

Para el diseño de la base de datos se utilizó el modelo EAV (entidad-atributo-valor) también conocido como Objeto-Atributo-Valor o Esquema Abierto, se usa en casos donde el número de atributos (propiedades, parámetros) usados para describir una entidad u objeto es potencialmente grande, pero que aplicados individualmente a una entidad concreta es pequeño. Las circunstancias donde se suelen aplicar son:

- Entidades con atributos heterogéneos:
- ✓ Entidades dinámicas en el tiempo, donde los atributos de una entidad son fijos, pero durante un período de tiempo variable. Es decir, son cambiantes (pueden crecer o decrecer) en el tiempo.

- ✓ Entidades conceptualmente dinámicas, dependiendo de la interpretación del sistema en un determinado momento. Es decir, entidades cuyo número de atributos cambia según el contexto (por ejemplo: atributos definidos por el usuario)
- Entidades con atributos homogéneos muy poco densos. Es decir, entidades con una desproporción muy grande entre el número de atributos posibles y el número de atributos con valor (no nulos).

El caso objeto de estudio consiste precisamente en entidades conceptualmente dinámicas, cuyos atributos cambian según el contexto y que en dependencia de esto son definidas por el usuario.

Este modelo se suele resolver sobre el modelo relacional con tres tablas: la de entidad, la de atributos posibles de dicha entidad y la de valores. La información se registra conceptualmente en la tabla de valores, que relaciona la información a través de tres columnas: clave ajena de entidad, clave ajena de atributo y valor.

En este caso la tabla entidad sería `dat_caso` y `dat_basecaso`, la tabla atributo sería `dat_atributo` y la tabla valor sería `dat_valordominiocelda`.

2.4.3 DISEÑO DE LOS PROCEDIMIENTOS

2.4.3.1 DIAGRAMA DE CLASES DEL DISEÑO WEB.

Los diagramas de clases son ampliamente utilizados en el modelado de sistemas orientados a objetos y se emplean para representar las relaciones que se establecen entre las clases. Se utilizan para modelar la vista de diseño estática de un sistema. En este subepígrafe se muestra el diagrama correspondiente al requisito funcional Gestionar Base de Casos, los otros diagramas se encuentran en los anexos del documento.

Diagrama de Clase del Diseño – RF Gestionar Base de Casos

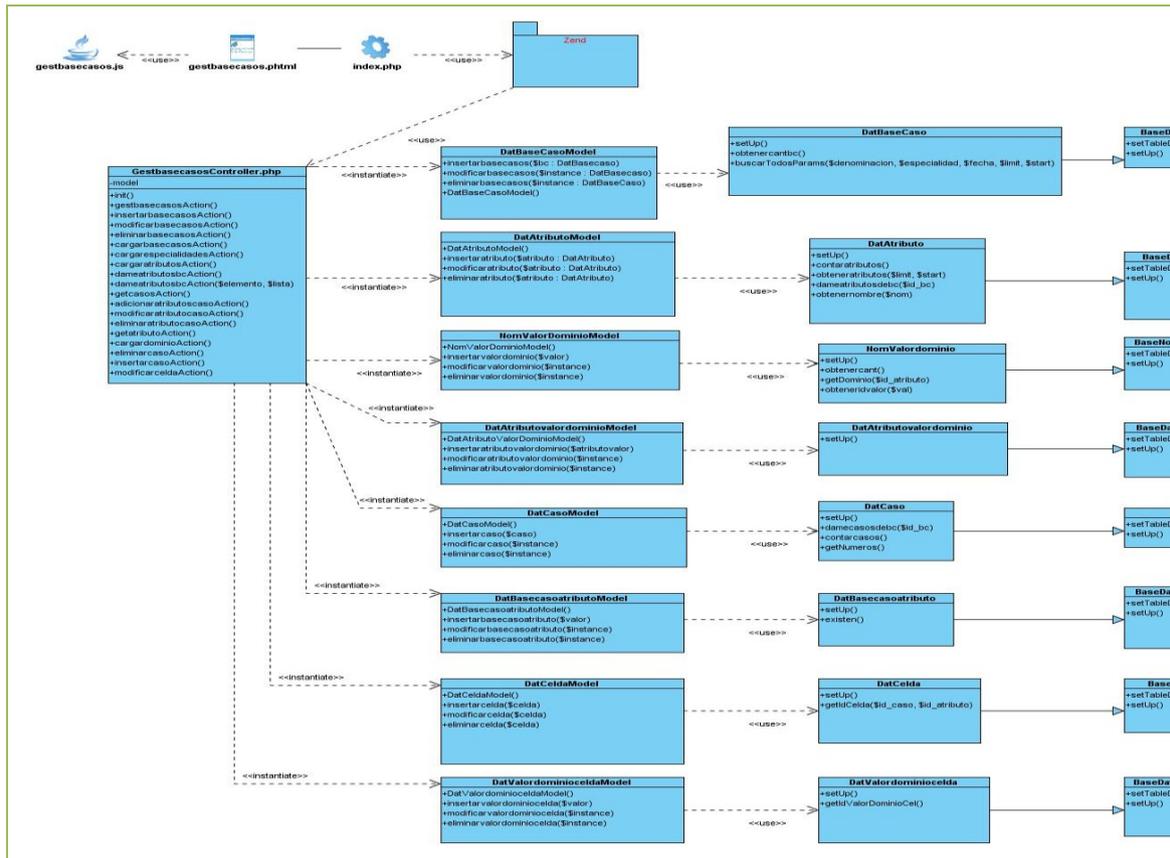


Fig.20: Diagrama de Clase Gestionar Base de Casos (Ver Anexos)

Este diagrama permite tener una descripción detallada de todas las clases del diseño, los principales atributos que posee y sus principales métodos. En esta figura se muestra el diagrama para una clase, las otras se pueden ver en los **Anexos**.

2.4.3.2 DIAGRAMA DE SECUENCIA

Los diagramas de interacción se utilizan para modelar la vista dinámica del diseño, en ellos se muestran las relaciones que se establecen entre los objetos y los diferentes mensajes que se pueden enviar entre ellos.

A continuación se muestra un ejemplo de los diagramas de secuencia, para más información consultar **Anexo 4**.

Diagrama de Secuencia – RF Gestionar Base de Casos – Escenario Adicionar Base de Casos

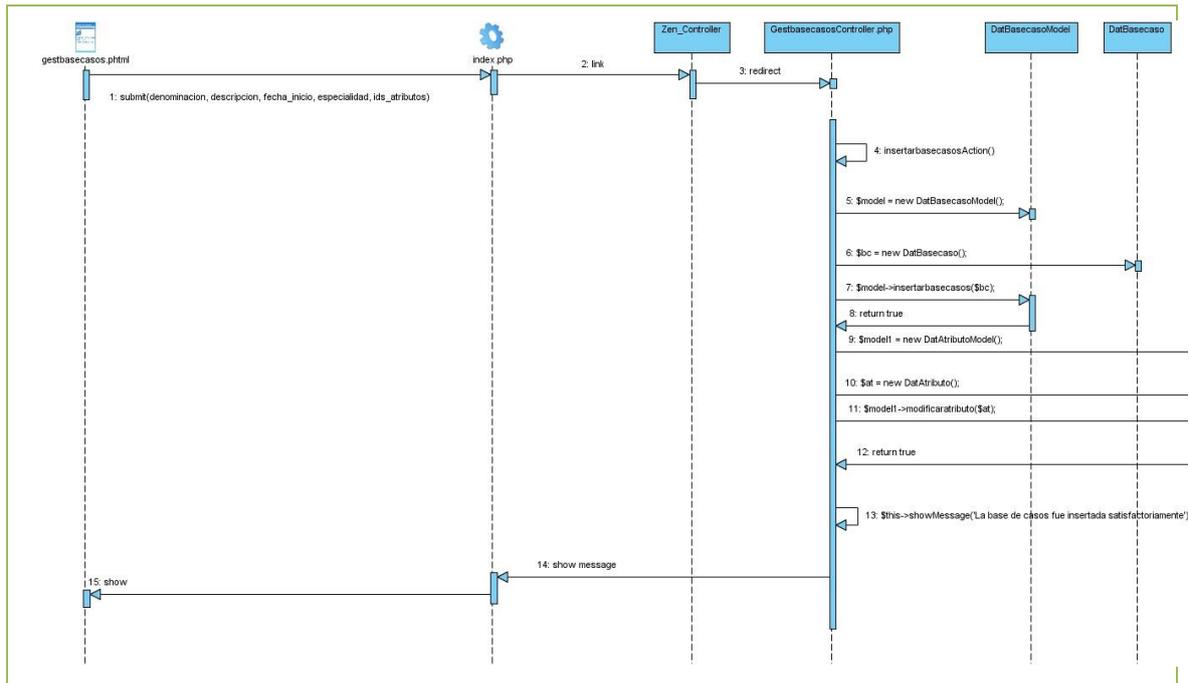


Fig.21: Diagrama de Secuencia Adicionar Base de Casos (Ver Anexos)

2.5 DISEÑO DE LAS INTERFACES

En este epígrafe se muestran algunas de las interfaces que se han diseñado para el sistema, cada una de ellas permite la interacción del usuario con la aplicación.

Los prototipos de interfaz de usuario son una representación que contienen la apariencia y distribución funcional que puede tener en el futuro una aplicación que se desarrolla, por tal motivo contribuyen al entendimiento entre desarrolladores y clientes así como a la satisfacción de estos últimos.

Para su esbozo y ejecución se tuvieron en cuenta los estándares definidos por el centro y fueron desarrollados sobre el framework para la capa de presentación Ext js.

Interfaz RF Adicionar Base de Conocimiento

Adicionar Base de Conocimiento

Características Atributos

Denominación:

Descripción:

Tahoma B I U A A ab

Fecha Inicio:

Especialidad: Seleccione la especialidad...

Cancelar Aplicar Aceptar

Fig.22: Interfaz RF Adicionar BC

Interfaz RF Gestionar Caso

Gestionar Base de Conocimiento

Adicionar Modificar Eliminar Ver detalles Ayuda

Denominación: Especialidad: Seleccione... Fecha de inicio: Buscar

Denominación	Descripción	Fecha Inicio	Especialidad
base 21	sddsvvs	2010-03-24	Logistca
base1	cvbcvb	2010-03-16	Seleccione la especialidad...
base6	la diva de bollo manso	2010-03-05	Inteligencia Militar
base7	dsfcvsd	2010-03-03	Logistca
base8	dsfcvsd	2010-03-03	Logistca
base9	simem	2010-03-23	Infanteria
base10	qwdad	2010-03-11	Logistca

Página 1 de 3 Resultados de 1 - 22 de 22

Base de Casos

Atributos Casos

- Adicionar
- Eliminar

Página 1 de 3 Resultados de 1 - 22 de 22

Fig.23: Interfaz RF Gestionar Caso

Interfaz RF Gestionar Atributo y Gestionar Dominio

Adicionar Atributo

Atributo

Nombre:

Tipo:
Seleccione...

Dominio del Atributo

+ Añadir Valor - Eliminar

Valores

Cancelar Aplicar Aceptar

Fig.24: Interfaz RF Gestionar Atributo y Gestionar Dominio

Interfaz RF Inferir Decisión

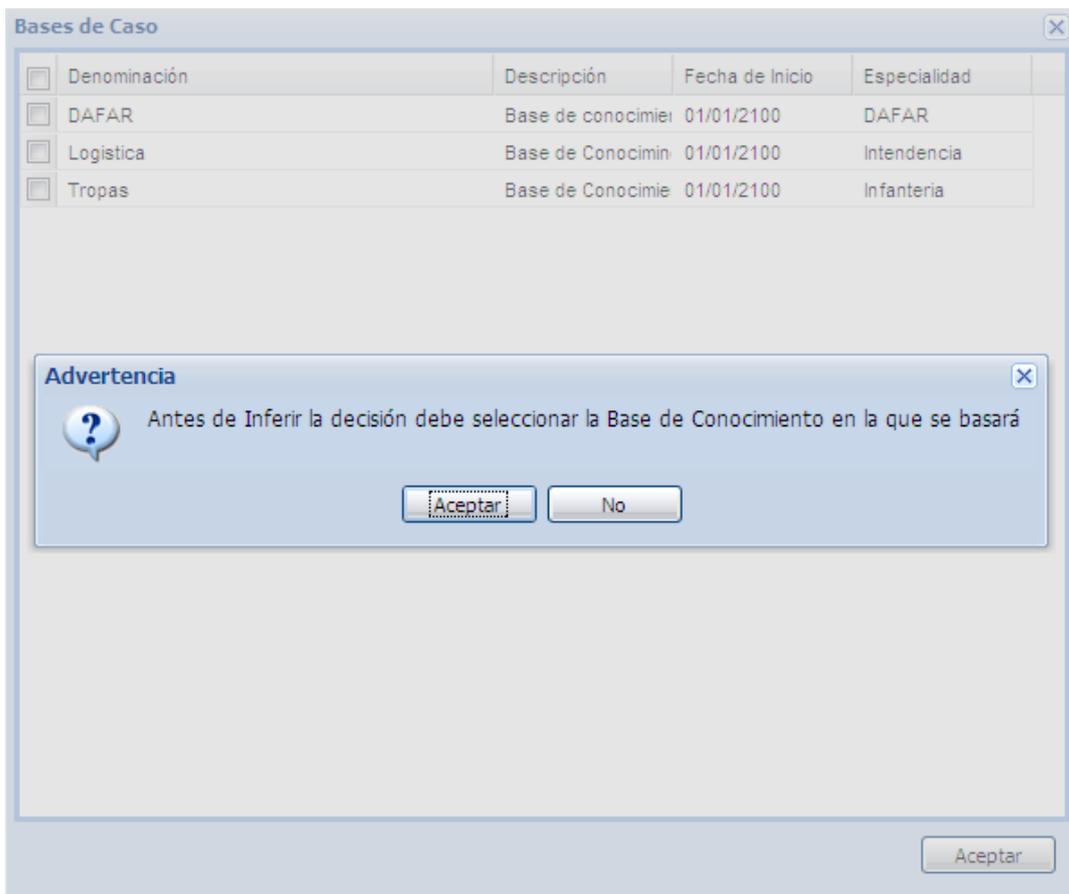


Fig.25: Inferir Decisión

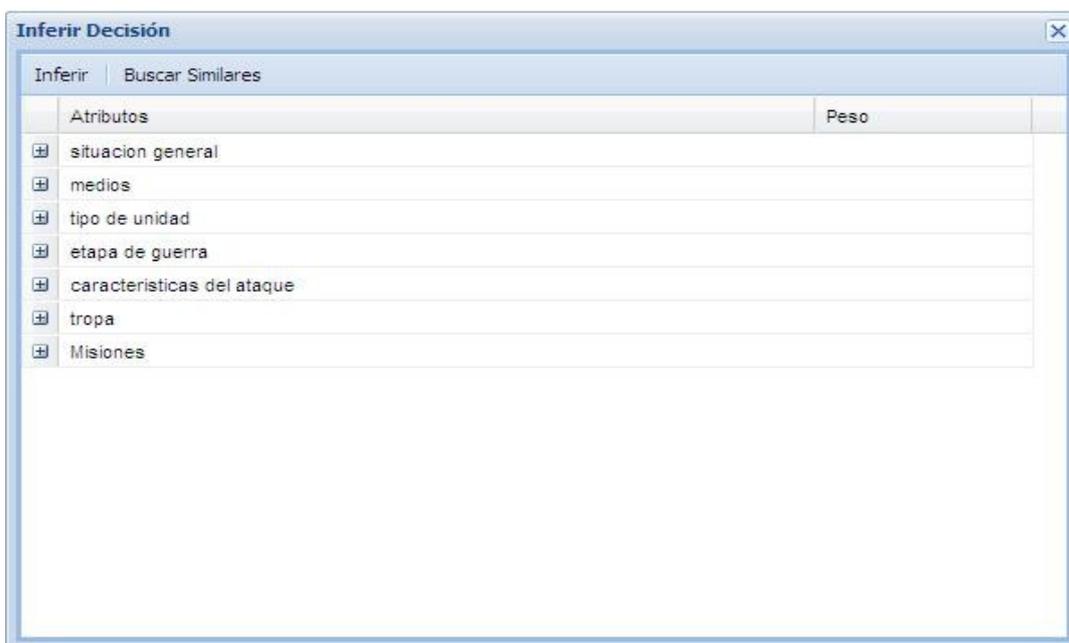


Fig.26: Inferir Decisión

2.6 Conclusiones Parciales

En el capítulo que concluye quedaron bien definidas cada una de las pautas necesarias para llevar a cabo la implementación de la solución, de esta manera queda claro que se va a hacer y como se va a hacer.

CAPITULO 3: VALIDACION DE LA SOLUCION

3.1 Introducción

En este capítulo se lleva a cabo la validación de la solución propuesta haciendo un análisis de los resultados que arrojó el sistema después de haber sido probado con un conjunto de bases de conocimientos y puesto en explotación experimental en la academia de las FAR, donde los usuarios principales tuvieron la posibilidad de interactuar con la aplicación y someterla a un conjunto de pruebas. Luego se muestra la evaluación realizada del sistema por un conjunto de expertos, los que fueron seleccionados a través del método Delphi.

3.2 Prueba contra bases de datos

Para hacer las pruebas a la aplicación se utilizaron las siguientes bases de datos internacionales:

1. Heart suministrada por el European Statlog project, Dept. Statistics and Modeling Science, Strathclyde University
2. Thyroids suministrada por el Garvan Institute of Medical Research, Sydney
3. Breast Cancer suministrada por W.H. Wolberg from the University of Wisconsin Hospitals, Madison [47]

Todas publicadas en el repositorio de aprendizaje automático de la Universidad de California UCI Repository of Machine Learning data bases [ver Anexo].

Para realizar la evaluación con las bases de datos se utilizó **El método de probabilidad de clasificación correcta**, para ello primero se particionó cada base de datos en dos, una parte de entrenamiento y otra de prueba, se seleccionaron 18 casos de los contenidos en la parte de pruebas y se sometieron a ensayo.

		Clase Verdadera C	
		0(+)	1(-)
Clase predicha por el modelo de clasificación C_m	0(+)	a	b
	1(-)	c	d

Tabla 6: Matriz de clasificación

En la tabla 6, la variable C_m recoge los resultados obtenidos con el modelo clasificatorio. Las cuatro letras incluidas en la tabla se interpretan de la siguiente manera:

- a es el número de casos en los que la clase verdadera es 0 y el modelo clasificatorio predice también 0
- d es el número de casos en los que la clase verdadera es 1 y el modelo clasificatorio predice también 1
- c es el número de casos en los que la clase verdadera es 0 y el modelo clasificatorio predice 1
- b es el número de casos en los que la clase verdadera es 1 y el modelo clasificatorio predice 0

Por tanto en $a + d$ casos la clasificación es correcta, mientras que en $c + b$ casos se han cometido errores en la clasificación. Definimos

- tasa de acierto = $\frac{a + d}{a + b + c + d}$
- tasa de error = $\frac{c + b}{a + b + c + d}$

De acuerdo a lo anterior, se puede aseverar:

Si $\text{tasa de acierto} > \text{tasa de error}$

Entonces se puede plantear que está correcto el método de clasificación que se utiliza.

De los 18 casos seleccionados

$$a = 7 \quad d = 5 \quad c = 2 \quad b = 2$$

$$\text{Tasa de acierto} = 0.66 \quad \text{y tasa de error} = .022$$

Como $\text{tasa de acierto} > \text{tasa de error}$

Entonces estadísticamente el clasificador es correcto

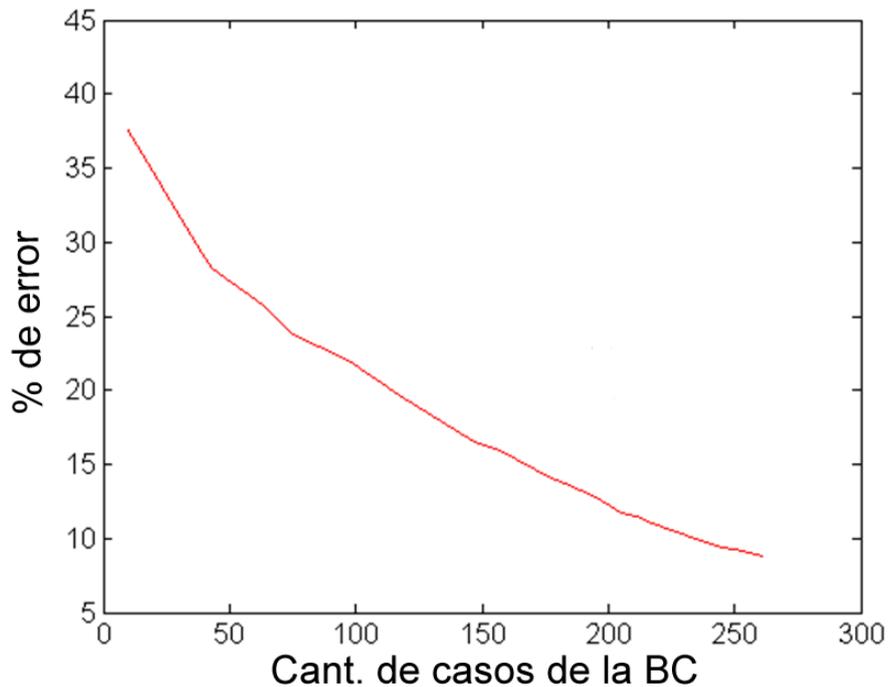


Fig. 2 : Muestra el por ciento de error contra la cantidad de casos de la BC

Mientras mayor cantidad de casos existan menos probabilidad tendrá el clasificador de equivocarse, por eso siempre es factible introducir en la base de casos no menos de 200 casos de forma tal que el sistema pueda comparar con mayor número de elementos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1		0,20	0,20	0,20	0,20	0,50	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,50	0,20	0,20	0,50
2			0,50	0,00	0,00	0,50	0,20	0,20	0,00	0,20	0,20	0,20	0,00	0,00	0,00	0,00	0,00	0,20	0,20	0,00	0,00	0,20
3				0,00	0,00	0,20	0,00	0,00	0,20	0,20	0,00	0,00	0,00	0,20	0,20	0,00	0,00	0,00	0,20	0,00	0,00	0,00
4					0,20	0,20	0,50	0,20	0,20	0,00	0,00	0,20	0,00	0,20	0,50	0,00	0,00	0,00	0,00	0,20	0,00	0,00
5						0,20	0,50	0,20	0,00	0,20	0,00	0,20	0,00	0,00	0,20	0,00	0,00	0,00	0,00	0,20	0,20	0,20
6							0,50	0,50	0,20	0,50	0,20	0,20	0,20	0,00	0,00	0,20	0,20	0,20	0,20	0,00	0,20	0,20
7								0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,50	0,50	0,50
8									0,80	0,80	0,20	0,20	0,20	0,20	0,20	0,50	0,20	0,20	0,20	0,80	0,20	0,20
9										0,80	0,20	0,20	0,20	0,00	0,20	0,80	0,80	0,50	0,50	0,20	0,20	0,20
10											0,20	0,20	0,20	0,20	0,20	0,50	0,50	0,20	0,20	0,20	0,20	0,20
11												0,20	0,00	0,00	0,00	0,20	0,20	0,50	0,20	0,00	0,00	0,00
12													0,20	0,20	0,20	0,20	0,20	0,20	0,00	0,20	0,50	0,20
13														0,20	0,00	0,00	0,00	0,00	0,00	0,20	0,20	0,20
14															0,00	0,20	0,20	0,20	0,00	0,00	0,20	0,20
15																0,00	0,00	0,00	0,00	0,00	0,00	0,20
16																	0,80	0,20	0,50	0,50	0,00	0,20
17																		0,20	0,20	0,20	0,00	0,00
18																			0,20	0,20	0,00	0,00
19																				0,20	0,00	0,00
20																					0,00	0,00
21																						0,20
22																						

Tabla 7: Matriz de semejanza entre casos probados para evaluar las funcionalidades del sistema

3.3 Criterio de expertos. Implementación del método Delphi.

Para llevar a cabo la validación de la solución propuesta se utilizó el método de expertos Delphi. Se seleccionaron 9 posibles expertos y se evaluaron sus conocimientos y habilidades relacionadas con:

- Inteligencia Artificial.
- Razonamiento Basado en Casos.
- Algoritmos de semejanza.

Se llevaron a cabo dos rondas de preguntas y se estableció el coeficiente de competencia para cada posible experto a partir de la fórmula: $K = \frac{1}{2} (kc + ka)$ Donde: **K**: Coeficiente de competencia de cada experto.

kc: Coeficiente de conocimiento o información que tiene el experto acerca del problema, calculado sobre la valoración del propio experto en una escala del 0 al 10 y multiplicado por 0,1. De esta forma, la evaluación "0" indica que éste no tiene absolutamente ningún conocimiento de la problemática correspondiente, mientras que la evaluación "10" significa que el experto tiene pleno conocimiento de la problemática tratada. Entre estas dos evaluaciones extremas hay nueve intermedias. El experto deberá marcar con una cruz en la casilla que estime pertinente.

ka: Coeficiente de argumentación o fundamentación de los criterios de el experto, obtenido como resultado de la suma de los puntos alcanzados a partir de una tabla patrón como se muestra en la siguiente tabla.

Tabla 8: Elementos a tener en cuenta para seleccionar el coeficiente de competencia de los expertos.

Fuentes de argumentación	Grado de influencia de cada una de las fuentes en sus criterios.		
	Alto(A)	Medio(M)	Bajo(B)
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores nacionales	0.05	0.05	0.05
Trabajos de autores extranjeros	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05

El coeficiente de competencia para cada experto se muestra en la tabla siguiente

Experto	Coeficiente de Competencia	Coeficiente alto Si $0.8 < K < 1.0$	Coeficiente medio si $0.5 < K < 0.8$	Coeficiente bajo Si $K < 0.5$
1	0.95	X		
2	0.87	X		
3	0.95	X		
4	0.85	X		
5	0.85	X		
6	0.35			X
7	0.95	X		
8	0.95	X		
9	0.67		X	

Todos los coeficientes de los posibles expertos se encuentran por encima de 0.25 que es el rango mínimo necesario, pero a pesar de ello se decidió escoger solo aquellos cuyo coeficiente se encontraba por encima de 0.8 y con una tendencia al rango máximo que es unidad. Además del conocimiento de los expertos también se tuvo en cuenta para su selección su posibilidad real de participación. Finalmente los expertos escogidos fueron 7 y todos de una forma u otra están vinculados con la inteligencia artificial. En cuanto a su grado científico 3 son doctores con más 15 años de experiencia y el resto maestros.

El cuestionario elaborado consta de 5 preguntas las cuales están encaminadas a evaluar la importancia del sistema, así como el cumplimiento de los requisitos del mismo.

Las respuestas a estas preguntas han sido categorizadas en: muy adecuado(C1), bastante adecuado (C2), adecuado (C3), poco adecuado (C4), no adecuado (C5).

Finalmente en la última pregunta se solicita una valoración general sobre la misma. El cuestionario fue enviado por correo electrónico o se les entregó personalmente en copia a dura a todos los expertos, explicándoles las características del método y la finalidad del mismo.

3.4 Análisis De Los Resultados

3.5 Conclusiones Parciales

Teniendo en cuenta la valoración estadística y de expertos anteriormente realizada se validó la propuesta del sistema y con ello la implementación del razonador que utiliza el software para la inferencia. Lográndose así un producto terminado y completamente validado, listo para usarse como apoyo al proceso de toma de decisión de los jefes y como método didáctico que tributará al proceso de enseñanza y

aprendizaje de las diferentes asignaturas de mando y dirección en la academia de las FAR.

Conclusiones

El manejo de un sistema de apoyo a la toma de decisiones reporta a la entidad una serie de beneficios que consisten, por ejemplo en: el incremento en la velocidad del procesamiento de datos, aumento en la productividad, ahorro de tiempo, mejora la calidad en la toma de decisiones lo que reduce los costos a lo largo de toda la organización, así como brinda una gran capacidad de almacenamiento de información y el procesamiento de datos.

Como se puede observar a lo largo de este trabajo, resulta imprescindible para las empresas en la actualidad contar con el soporte tecnológico que les permita almacenar de manera centralizada toda la información que utilizan y de esta forma tomar mejores decisiones.

Como la implementación de estas herramientas resulta costosa y requiere amplios períodos de tiempo para que los expertos completen las bases de conocimientos, una vez que las conformen hagan salvas para no perder la información.

Gracias a las Bases de conocimientos, la experiencia se mantiene vigente y pasa de generación en generación aunque las personas que las vivieron ya no estén en la entidad.

Aunque para estas tareas los Sistemas Inteligentes todavía no sean 100% autónomos, resultan un apoyo vital, por sus características específicas, ya que son capaces de procesar grandes cantidades de información y extraer patrones tanto evidentes como no evidentes; por otra parte, otro punto a su favor es la falta de prejuicios y suposiciones, que son algunas de las principales causas de fallos humanos.

Por ahora las aplicaciones de los Sistemas Inteligentes en esta esfera son instrumentos de apoyo a la toma de decisiones por parte de los humanos, pero al igual que está ocurriendo con las aplicaciones dentro de los campos de batalla, la tendencia es a que estos sistemas cada vez tengan más autonomía. Seguramente, en un futuro relativamente cercano, estos sistemas tendrán una importante capacidad de decisión y serán piezas fundamentales que nos protejan a diario de las importantes amenazas a las que nos hemos visto sometidos, especialmente, en los últimos años.

Reconocemos el valor y la importancia de la tecnología, pero no olvidemos que el factor humano siempre será insustituible.

Bibliografía Referenciadas:

1. <http://www.eumed.net/ce/2006/mcss.htm>
2. <http://www.monografias.com/trabajos16/toma-de-decisiones/toma-de-decisiones.shtml>
(25 de septiembre de 2009)
3. <http://www.monografias.com/trabajos60/guerra-ciencia-tecnologia/guerra-ciencia-tecnologia2.shtml> (1 de octubre 2009)
4. <http://www.ara.mil.ar/Publicacion%20696/03.pdf> (1 de octubre 2009)
5. <http://www.grups.pangea.org/pipermail/infomoc/Week-of-Mon-20011015/000076.html>
(2 de octubre 2009)
6. <http://www.geocities.com/pasteda/armas2.htm> (2 de octubre 2009) 3:02 pm
7. <http://www.escuelasuperiordeguerra.iese.edu.ar/adiestrador.htm> (2 de octubre 2009)
3:20 pm
8. <http://ri.conicyt.cl/575/article-10953.html> (2 de octubre 2009) 3:31 pm
9. <http://www.monografias.com/trabajos16/resistencia-al-cambio/resistencia-al-cambio.shtml> (5 de octubre 2009) 12:20pm
10. http://weblogs.madrimasd.org/sistemas_inteligentes/archive/2009/07/07/121393.aspx
(5 de octubre 2009) 3:18 pm
11. Turban, Efraim y Aronson Jay E. (2001), Decision Support Systems And Intelligent Systems, Sixth edition, Prentice Hall, New Jersey.
12. http://bvs.sld.cu/revistas/aci/vol18_3_08/aci03908.htm
13. OJO, importante <http://www.eumed.net/ce/2006/mcss.htm>
14. (Martín del Brío & Sanz Molina)
15. <http://www.invenia.es/oai:dialnet.unirioja.es:ART0000010968>
16. http://www.cio.mx/3_enc_mujer/files/extensos/Sesion4/S4-ING10.doc
17. http://www.wikilearning.com/monografia/propuesta_de_clasificacion_de_las_herramientas_software_para_la_gestion_del_conocimiento-el_software_para_la_gc/7727-19
18. <http://www.eumed.net/ce/2009a/arrs.htm>
19. http://www.uci.cu/files/investigaciones/RCCI/RCCI_Vol_1_No_3.pdf
20. <http://www.microsoft.com/conosur/hechos/studies/fuerzaaerea.aspx>
21. <http://www.microsoft.com/conosur/hechos/studies/fuerzaaerea.aspx>
22. http://www.acapomil.cl/investigacion/boletines/boletin_2001/articulos/articulo_05.htm
23. http://www.icrc.org/WEB/SPA/sitespa0.nsf/html/simulacion_El_Salvador_2005
24. http://www.inteligenciaartificial.cl/ciencia/software/ia/inteligencia_artificial.htm
25. <http://www.alegsa.com.ar/Dic/inteligencia%20artificial.php>
26. <http://www.mastermagazine.info/termino/5394.php>
27. <http://www.answermath.com/inteligencia-artificial.htm>

28. http://www.informatica2007.sld.cu/Members/ricardo_fernandez/razonamiento-basado-en-casos-en-ciencias-medicas-sobre-plataforma-web-1/
29. <http://www.iie.org.mx/gee/arti/21.pdf>
30. http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V03-4THC1BV-7&_user=2342189&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_searchStrId=1103377474&_rerunOrigin=google&_acct=C000056883&_version=1&_urlVersion=0&_userid=2342189&md5=54f95a7f9adbec3c58832db1ba59436 (interfaces para CBR)
31. <http://www.monografias.com/trabajos10/intelart/intelart.shtml#defia>
32. <http://www.monografias.com/trabajos16/la-inteligencia-artificial/la-inteligencia-artificial.shtml>
33. Sosa Sierra, María del Carmen. Introducción a las técnicas de inteligencia artificial aplicadas a la gestión financiera empresarial. Eumed.net. [En línea] [Citado el: 05 de Abril de 2010.] <http://www.eumed.net/ce/2006/mcss.htm>.
34. Carlos Soto, Marlene. SISTEMA EXPERTO DE DIAGNOSTICO MEDICO DEL SÍNDROME DE GUILLIAN BARRE. s.l. : UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS, 2002.
35. <http://techtastico.com/post/zend-framework-una-introduccion/>
36. Requerimientos y versiones de las tecnologías y herramientas del marco de trabajo Sauxe 1.5.4 Beta.
37. Kolodner. J, Leaked. D.A tutorial introduction to case based reasoning. Case based reasoning: Experiences, Lesson and future directions, Menlo Park AAAI Press/MIT (1996).
38. Agudo, B. Introducción al Razonamiento Basado en Casos. Departamento de Sistemas informáticos y programación. Facultad de Matemáticas. Universidad Complutense de Madrid. Madrid. España (2001).

Bibliografía consultada

1. <http://books.google.com/cu/books?id=D9sXTQGhID0C&pg=PT214&lpg=PA213&ots=-10ngxNG55&dq=codigo+motor+de+inferencia+en+java#v=onepage&q=&f=false>
(Diseño y aplicación de modelos multiagente para la ayuda a la decisión)
2. <http://salaboy.wordpress.com/category/java/>
(ide para la realización de motores de inferencia)
<http://www.iiia.csic.es/udt/es/blog/jrodriguez/2009/razonamiento-basado-en-casos-o-case-based-reasoning-cbr>(habla del razonamiento basado en casos. Estructura)

https://www.cenitsegura.com/cenit/index.php?option=com_acajoom&act=mailing&task=view&listid=1&mailingid=27&Itemid=128

<http://www.revistaciencias.com/publicaciones/EEuFVykIVuYpTtivZQ.php>

<http://www.monografias.com/trabajos39/aplicaciones-inteligencia-artificial/aplicaciones-inteligencia-artificial2.shtml>