

Universidad de las Ciencias Informáticas



Facultad 5

Título: Propuesta de aplicación de Inteligencia Artificial para el tratamiento de las alarmas del SCADA.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor(es): Dagmara Santiesteban Mecias.

Geomar Lazaro Benitez Sierra.

Tutor : Ing. Yaima Antúnez Ojeda.

Co-Tutor: Ing. Adiel Duran Rodríguez.

Ciudad de La Habana, junio 2011

“Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los __ días del mes de _____ del año _____.

Dagmara Santiesteban Mecias.

Autor

Geomar Lazaro Benitez Sierra.

Autor

Ing. Yaima Antúnez Ojeda.

Tutor

Ing. Adiel Duran Rodríguez.

Co-Tutor

DATOS DE CONTACTO

Tutor:

Nombre y Apellidos: Yaima Antúnez Ojeda.

Institución: Universidad de las Ciencias Informáticas (UCI).

E-mail: yantunez

Graduada de la UCI en el año 2008, Cinco años en la producción de software.

Co-Tutor

Nombre y Apellidos: Adiel Duran Rodríguez.

Institución: Universidad de las Ciencias Informáticas (UCI).

E-mail: aduranr

AGRADECIMIENTOS

**A mi mamá por ayudarme y apoyarme a lo largo de mi vida,
A mi papá por contar siempre con su cariño,
A Rubén por darnos su mano amiga incondicionalmente,
A Juan Miguel por guiarnos en los momentos de desesperación,
A nuestros tutores por su ayuda en la realización de esta tesis,
A Geomar por siempre estar a mi lado y ser mi soporte,
A todos ellos muchas gracias, de corazón.
Dagmara**

**A mi abuela y Abuelo por confiar en mí y darme su apoyo en todo momento,
A mi tío Pitio por darme todo el amor y cariño que un hijo puede tener,
A mi mamá por apoyarme y quererme en cada instante de mi vida,
A mi papá, que sin su ejemplo y fuerza brindada no sería hoy el hombre que soy,
A mi tía Olguita por quererme como a un hijo,
A Yanetsy por su cariño y comprensión,
Al Rubén por aguantarme en estos 5 años,
Al Flaco por no decir nunca que no, ante un pedido de ayuda,
A Hardy por su ayuda y guía en esta tesis,
Al compañero Bridon por sus consejos para esta tesis,
A mis tutores por su dedicación y ayuda,
A todos mis amigos por formar parte de mi familia en todo este tiempo,
A mi querida novia Dagmara, por estar conmigo en los buenos y malos momentos,
A todos ellos, y a todos que me ayudaron a que este día llegara muchas gracias.
Geomar**

DEDICATORIA

A mis padres, que esperaban este momento, en ver a su hija como una profesional, gracias por darme la fuerza para conseguirlo, este es mi regalo.

Dagmara.

A Mima y a Pitio, esto es una de las muchas cosas que gracias a ustedes he obtenido, por eso, se las dedico, porque sé que verme como ingeniero, es la alegría más grande que les puedo dar.

Geomar.

RESUMEN

Los Sistemas SCADA (Supervisory Control and Data Acquisition) son de gran utilidad en la actualidad ya que permite la recolección de información utilizada posteriormente por un operador para tomar decisiones sobre un sistema en específico. El sistema SCADA UX desarrollado en la Facultad 5 consta actualmente entre sus módulos con uno para el tratamiento de alarmas, en el cual, es necesario lograr un comportamiento inteligente, para que de esta forma, sea mucho más fácil su entendimiento y control por parte del operador. En este trabajo se propone dar solución a ese problema, realizando una investigación sobre las técnicas de Inteligencia Artificial, comparándolas entre si y viendo cuál de estas es más afín para el sistema de alarmas del propio SCADA. Para resolver dicho problema se propondrán los artefactos ingenieriles para la creación de un módulo de Inteligencia Artificial mediante el cual se trataran las alarmas del sistema y por último se mostrará una aplicación demostrativa que servirá de base en el desarrollo del futuro módulo.

PALABRAS CLAVE

Inteligencia Artificial, SCADA, Sistemas de Alarmas.

Índice de Contenido.

DATOS DE CONTACTO	II
AGRADECIMIENTOS	III
DEDICATORIA.....	IV
Capítulo 1: FUNDAMENTACIÓN TEÓRICA.	6
1.1 Inteligencia Artificial.	6
1.2. Técnicas de Inteligencia Artificial.	7
1.2.1 Sistemas Expertos.	7
1.2.2 Redes Neuronales Artificiales (RNA).	18
1.3 Los sistemas SCADA.	20
1.3.1 Ventajas del SCADA.	21
1.3.2 Subsistemas de SCADA.	21
1.3.3 Las Alarmas en el Sistema.	25
1.4 Selección de la Técnica a utilizar.	31
1.5 Herramientas y Metodologías utilizadas.	33
1.5.1 RUP como metodología de desarrollo.	33
1.5.2 Visual Paradigm.	33
1.5.3 UML como lenguaje de modelado.	33
Capítulo 2: PROPUESTA DE SOLUCIÓN.	35
2.1 Reglas del Negocio.....	35
2.2 Captura de requisitos.....	36
2.2.1 Requisitos funcionales.....	36
2.2.2 Requisitos no funcionales.	37
2.3 Conexión con el scada.....	39
2.4 Propuesta del sistema.	39

2.5 Descripción de la base de conocimiento.	40
2.5.1 El Proceso de Razonamiento.	41
2.5.2 Algoritmo de Inferencia.	42
2.6 Modelo de Casos de Uso del Sistema.	42
2.6.1 Actores del sistema.	42
2.6.2 Casos de uso del sistema.....	43
2.6.3 Diagrama de casos de uso (CU) del sistema.....	43
2.6.4 Descripción de los casos de uso en formato extendido.....	43
2.6.5 Diagramas de clases.	47
Capítulo 3: VALIDACIÓN	51
3.1 Diagrama de Componentes.	52
3.2 Formularios de la aplicación.	52
3.3 Diseño de casos de Prueba.	60
3.3.1 Diseño de caso de Prueba.CU Gestionar Regla.	60
3.3.2 Diseño de caso de Prueba.CU Generar Respuestas de Alarma.	61
CONCLUSIONES	63
RECOMENDACIONES	64
BIBLIOGRAFÍA	65
GLOSARIO	68

Índice de Figuras.

<i>Fig. 1 Arquitectura de un SBR.</i>	9
<i>Fig. 2 Separación Direccional</i>	16
<i>Fig. 3 Arquitectura General del SCADA</i>	22
<i>Fig. 4 Diagrama de CU.</i>	43
<i>Fig. 5 Diagrama de Clases de Análisis. Gestionar Regla.</i>	47
<i>Fig. 6 Diagrama de Clases de Análisis. Generar Respuesta de Alarma.</i>	48
<i>Fig. 7 Diagrama de Clase del Diseño General.</i>	48
<i>Fig. 8 Diagrama de Secuencia. Adicionar Regla.</i>	49
<i>Fig. 9 Diagrama de Secuencia. Modificar Regla</i>	49
<i>Fig. 10 Diagrama de Secuencia. Eliminar Regla</i>	50
<i>Fig. 11 Diagrama de Secuencia. Generar Respuesta de Alarma.</i>	50
<i>Fig. 12 Diagrama de Componentes.</i>	52
<i>Fig. 13 Formulario Principal</i>	53
<i>Fig. 14 Formulario Adicionar.</i>	54
<i>Fig. 15 Regla Adicionada</i>	55
<i>Fig. 16 Selección de la Regla a Modificar.</i>	56
<i>Fig. 17 Formulario Modificar con la regla cargada.</i>	56
<i>Fig. 18 Selección de la regla a eliminar</i>	57
<i>Fig. 19 Regla eliminada de la Base de Conocimientos.</i>	58
<i>Fig. 20. Sumario sin Respuestas de Alarmas.</i>	59
<i>Fig. 21. Sumario con Respuesta de Alarmas.</i>	59

Índice de Tablas

<i>Tabla 1. Actores del Sistema.</i>	43
<i>Tabla 2. Descripción del CU Generar Respuestas de Alarma.</i>	45
<i>Tabla 3. Descripción del CU Gestionar Reglas.</i>	47
<i>Tabla 4 Diseño de Caso de Prueba. CU Gestionar Regla.</i>	61
<i>Tabla 5 Diseño de Caso de Prueba. CU Generar Respuesta de Alarma.</i>	62

INTRODUCCIÓN

No siempre se toma la decisión correcta en disímiles momentos de la vida, ya sea por un motivo o por otro, aunque nos damos cuenta de ello al pasar el tiempo o cuando ya es tarde. Para la toma de decisiones es necesario conocer, comprender y analizar el problema, sea cual sea su índole, después de esto, mediante la reflexión y analizando los datos guardados referentes a la situación actual vemos cual es el camino o solución a tomar. El surgimiento de las computadoras brindó la posibilidad de crear disímiles programas que de una forma u otra facilitan el trabajo, automatizando servicios y operaciones por medio de los sistemas computarizados. Después de esto, el hombre empezó a trabajar en base a crear programas mediante los cuales las máquinas desarrollen comportamientos humanos, de esta forma se origina el campo de la Inteligencia Artificial (IA). La cual comenzó a nutrirse de técnicas que, dentro de sus funciones principales, está la de brindar soporte para la solución de problemas referentes a la toma de decisiones.

En la Facultad 5 se desarrolla un sistema de SCADA, acrónimo de *Supervisory Control And Data Acquisition* (en español, Control Supervisor y Adquisición de Datos), estos sistemas de control de producción, se comunican con los dispositivos de campo y controlan el proceso de forma automática desde un ordenador. Proporcionan información del proceso con el fin de lograr una mayor retroalimentación sobre el operador o sobre el propio proceso.

Actualmente este sistema carece de un comportamiento y respuesta inteligente a la hora de gestionar las alarmas que se generan en él; esto trae desventajas para el sistema, tales como:

- El sistema SCADA solo reconoce los eventos anormales, no brinda una solución al problema dado.
- La pérdida de tiempo por parte del operador a la hora de la toma de decisiones con respecto a los problemas que presente en sistema.
- Escases de información a la hora de establecer la prioridad con que deben ser solucionados los problemas en el sistema.

Por los problemas antes planteados se establece como **problema científico**:

¿Cómo lograr un comportamiento inteligente en el tratamiento de las alarmas del sistema SCADA UX?

El **objeto de estudio** se enfoca en: Las técnicas de IA para la toma de decisiones, donde el **campo de acción** estaría delimitado al comportamiento inteligente para la toma de decisiones en el tratamiento de alarmas del sistema SCADA UX.

Con el fin de resolver el problema científico se trazó el siguiente **objetivo**:

Desarrollar un módulo de IA para la toma de decisiones en el tratamiento de alarmas del sistema SCADA UX.

Para darle solución a este problema, se desarrollan las siguientes **tareas investigativas**:

- Caracterización de las técnicas de IA para la toma de decisiones.
- Descripción del funcionamiento de las alarmas en el sistema SCADA UX.
- Selección de técnica de IA para la toma de decisiones del SCADA UX en el tratamiento de las alarmas.
- Selección de herramientas y metodologías a utilizar durante el desarrollo del módulo de IA para el tratamiento de alarmas.
- Realización del Análisis y Diseño del módulo de IA.
- Desarrollo de una aplicación demostrativa para la validación del módulo de IA.

Como **resultado** de este trabajo se pretende seleccionar la técnica de Inteligencia Artificial a utilizar en el módulo y brindar una propuesta del módulo para la toma de decisiones en el tratamiento de las alarmas del Sistema SCADA UX.

Para facilitar el alcance de estos resultados y una mayor organización se empleará los siguientes métodos científicos de investigación:

Métodos teóricos.

➤ **Analítico-Sintético:** Permite hacer un análisis independiente y diferencial sobre las distintas técnicas de Inteligencia Artificial, que posibilitará descubrir sus características generales, así como su relación y aplicación para resolver el problemas en cuestión e impulsar el desarrollo de la investigación.

➤ **Inducción-Deducción:** Utilizado para la revisión, estudio y justificación de la técnica seleccionada para el sistema de alarma de SCADA UX. Por medio de análisis individuales de las técnicas se llega a una propuesta general según las características de estas para seleccionar la más adecuada.

➤ **Histórico-Lógico:** Para conocer, con mayor profundidad, los antecedentes y las tendencias actuales referidas a cada una de las técnicas de IA, así como su aplicación en el área de los sistemas de alarmas.

Métodos empíricos.

➤ **Observación:** Mediante la observación se pueden analizar los resultados e investigar cómo se comporta el sistema de alarma.

El presente documento está compuesto por tres capítulos, estructurados de la siguiente manera:

Capítulo 1: "Fundamentación Teórica", ofrece un marco teórico profundizando en los principales conceptos de la investigación. Además de la selección de la técnica de IA a utilizar así como las herramientas y las metodologías que para el desarrollo del trabajo.

Capítulo 2: "Propuesta de solución", se exponen las características que va a tener el sistema. Se definen las reglas del negocio y el modelo de dominio, los requisitos funcionales y no funcionales. Se realiza el análisis del sistema que incluye los distintos diagramas de análisis.

Capítulo 3: "Validación del Módulo", se desarrolla un Aplicación demostrativa para la validación de la propuesta de solución.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

En este capítulo se brindan conceptos necesarios para la comprensión del trabajo, ya sean referentes a la IA, las técnicas que esta comprende, o sobre los sistemas SCADA, además se brinda información valiosa que será de suma importancia para el entendimiento del surgimiento de la propuesta de solución de este trabajo. Se seleccionarán también las herramientas y metodologías que posteriormente serán utilizadas en el desarrollo del trabajo y la técnica de IA a utilizar en el módulo para la gestión de alarmas del SCADA UX.

1.1 INTELIGENCIA ARTIFICIAL.

La Inteligencia Artificial se ha desarrollado como una de las ramas de la Ciencia de la Computación. El trabajo en la misma comenzó en la década de 1950. El término Inteligencia Artificial fue usado a partir de 1956.

La IA tiene múltiples definiciones pero de forma general 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 [\[1\]](#). Es la ciencia que se encarga de crear máquinas con capacidad de realizar funciones realizadas por personas que requieren de una conducta inteligente.

La IA es un campo del saber que ha experimentado un rápido desarrollo y se ha diversificado a partir de la década de 1990. El núcleo tradicional de la IA ha sido denominado IA simbólica tradicional o simbólica deductiva, y comprende el desarrollo de diversas formas de representación del conocimiento en forma simbólica y explícita y los métodos de solución de problemas usando este conocimiento basadas en técnicas de búsqueda, los sistemas basados en el conocimiento, el procesamiento de lenguaje natural, etc.

Nuevos campos se han desarrollado en esta disciplina, los cuales han potenciado la aplicabilidad de la misma. Entre ellos las redes neuronales artificiales, los algoritmos genéticos, los sistemas difusos

(fuzzysystems) y la Teoría de conjuntos rugosos (Rough Set Theory). Estos nuevos campos son denominados por algunos autores como computación blanda (Soft Computing).[\[2\]](#)

El empleo de la IA está orientado a aquellas profesiones que, ya sea por un motivo u otro necesitan disponer del conocimiento de un experto en el campo. Aquí es donde entra a trabajar la IA, ya que mediante esta se pueden obtener la disponibilidad a tiempo completo de un asistente artificial que solucione los problemas existentes sin cometer los errores propios del ser humano.

1.2. TÉCNICAS DE INTELIGENCIA ARTIFICIAL.

La IA se divide en dos ramas principales: la Inteligencia Artificial Convencional y la Inteligencia computacional.

En la IA convencional están los métodos conocidos como máquinas de aprendizaje, predominando en estos el análisis estadístico. Algunos métodos de esta rama incluyen: Sistemas expertos, Razonamiento basado en casos, Sistemas basados en Comportamiento y las Redes Bayesianas.

La Inteligencia Computacional va dirigida más al aprendizaje interactivo. Utilizando métodos como: Redes neuronales, Sistemas difusos, Computación evolutiva.

En el presente capítulo se tratan específicamente las técnicas más significativas empleadas en la solución de problemas correspondientes con la toma de decisiones. Ya que estas son de suma importancia para el desarrollo del presente trabajo.

1.2.1 Sistemas Expertos.

Es bastante común que los seres humanos tomen decisiones en condiciones de incertidumbre, es más, la incertidumbre hace parte del diario vivir de las personas, un ejemplo tan cotidiano como el de ir tarde al trabajo y preguntarse si tomar un bus o taxi, la persona decide con incertidumbre, porque no sabe si el bus se demore o no, o si estará lleno, por otro lado el taxi le costará más, pero debe tomar una decisión que lo lleve a su trabajo. Este tipo de decisiones se presenta a menudo tanto en la vida

cotidiana, como en el ámbito científico o en el empresarial; aunque la incertidumbre esté presente en todos esos campos, el ser humano ha desarrollado técnicas y teorías, como la probabilidad y estadística, para enfrentar este tipo de decisiones, pero además de estos métodos; el rápido desarrollo de los sistemas ha permitido fusionar todo el conocimiento en aplicaciones que permiten dar soluciones que aunque no son perfectas tienen un soporte teórico que le da mayor probabilidad de éxito, que a una decisión tomada al azar.

Un Sistema Experto puede ser definido como: el sistema informático que imita los procesos que realiza un experto en un área de especialidad, al resolver problemas y “debería ser capaz de procesar y memorizar información, aprender y razonar en situaciones deterministas e inciertas, comunicarse con los hombres y otros sistemas expertos, tomar decisiones apropiadas y explicar porque se han tomado tales decisiones”.

Los Sistemas Expertos pueden ser clasificados, de acuerdo al tipo de problemas con el que tratan, en dos grandes grupos: deterministas y estocásticos. Los sistemas basados en reglas son aquellos que tratan con problemas deterministas y los resuelven utilizando una serie de reglas de producción. Los sistemas probabilísticos son aquellos donde se utiliza la probabilidad como medida intuitiva de la incertidumbre.

Un Sistema Experto tiene básicamente dos partes: la base de conocimiento y la máquina de inferencia. La base de conocimiento corresponde a las afirmaciones de validez general tales como reglas, distribuciones de probabilidad, manuales, entre otros. La máquina de inferencia son los algoritmos que manejan la base de conocimiento para obtener conclusiones y propagar el conocimiento.

Los Sistemas Basados en Reglas (SBR) no tienen en cuenta la incertidumbre, ya que la forma de tratar los objetos y las reglas, es de tipo determinista, pero como se dijo anteriormente, es frecuente la toma de decisiones en condiciones de incertidumbre, ya sea en áreas de la ingeniería, en los negocios o inclusive en áreas tan sensibles como las sociales.

Los sistemas de tipo no deterministas, cuentan con una base de conocimiento, pero esta se forma por el espacio probabilístico, que describe el problema. La máquina de inferencias está basada en

probabilidades condicionales y se encarga de actualizar dichas probabilidades con base en los hechos que observa del ambiente en el que se desempeña.

De manera general existen tres tipos de sistemas expertos: (1) Basados en reglas previamente establecidas. (2) Basados en casos y (3) Basados en redes bayesianas. En cada uno de ellos correspondiente, la solución a un problema planteado se obtiene: (1) Aplicando reglas heurísticas apoyadas generalmente en lógica difusa para su evaluación y aplicación. (2) Aplicando el razonamiento basado en casos, donde la solución a un problema similar planteado con anterioridad se adapta al nuevo problema. (3) Aplicando redes bayesianas, basadas en estadística y el teorema de Bayes. [3]

1.2.1.1 Sistemas Basados en Reglas (SBR).

Un SBR es un programa de ordenador capaz de procesar información específica contenida en memoria, con un conjunto de reglas (contenidas en lo que se llamará base de conocimiento), mediante un motor de inferencia para deducir nueva información.

A continuación se muestra en la **Fig1** la Arquitectura de un SBR, para un mejor entendimiento de este sistema.

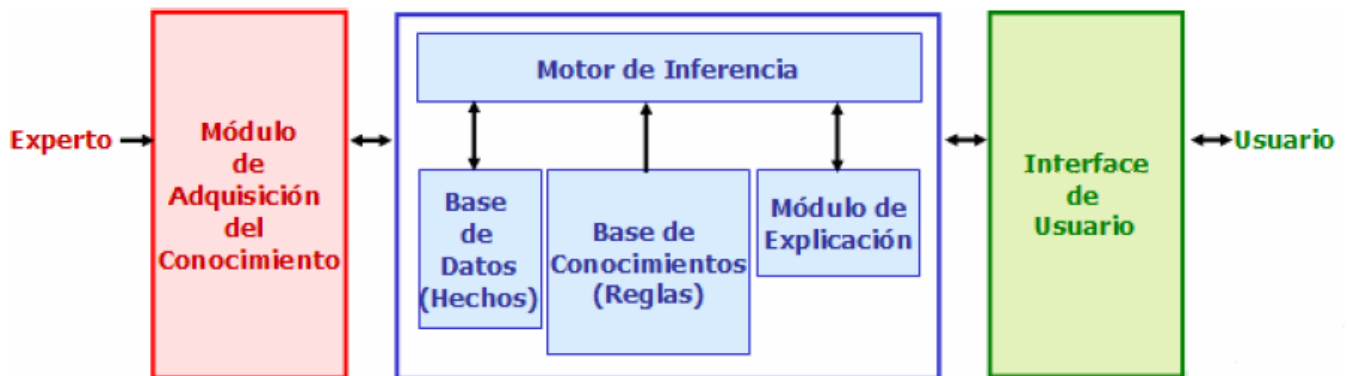


Fig. 1 Arquitectura de un SBR.

Mediante técnicas de búsqueda y procesos de unificación, los sistemas basados en reglas automatizan sus métodos de razonamiento y proporcionan una progresión lógica desde los datos iniciales, hasta las conclusiones deseadas. Esta progresión hace que se vayan conociendo nuevos hechos o descubriendo nuevas afirmaciones, a medida que va guiando hacia la solución del problema. [4]

LAS REGLAS EN LOS (SBR).

Las reglas utilizan un formato IF - THEN para representar el conocimiento, la parte IF de una regla es una condición (también llamada premisa o antecedente), y la parte THEN de la regla (también llamada acción, conclusión o consecuente) permite inferir un conjunto de hechos nuevos si se verifican las condiciones establecidas en la parte IF.

PROCESO DE RAZONAMIENTO.

El proceso de solución de problemas en un SBR es crear una cadena de inferencias que constituye un camino entre la definición del problema y su solución. Esta cadena de inferencias puede construirse por dos vías (direcciones de búsqueda):

- Comenzar con todos los datos conocidos y progresar hacia la conclusión (data driven o forward chaining).
- Seleccionar una conclusión posible y tratar de probar su validez buscando evidencias que la soporten (goal driven o backward chaining).

La dirección forward es apropiada cuando hay pocos datos de entrada o la cantidad de conclusiones posibles es grande. Entre los campos de aplicación para esta dirección de búsqueda está el monitoreo y diagnóstico en sistemas de control en tiempo real, diseño y planificación.

La dirección backward es apropiada cuando hay pocas conclusiones posibles o los valores de entrada no son adquiridos automáticamente. Los problemas de diagnóstico y clasificación son frecuentemente resueltos con esta dirección de búsqueda.

EXPLICACIONES EN LOS SBR.

El SBR debe ser capaz de ofrecer explicaciones al usuario cuando este se la pida. Usualmente hay dos momentos donde el usuario puede preguntar:

- Cuando se pide un nuevo dato el usuario.

En este caso el usuario puede querer saber por qué (**Why**) se le hace esa pregunta. Ocurre en el razonamiento backward.

- Cuando se termina el proceso de inferencia.

El usuario puede querer saber cómo (**How**) se alcanzó la respuesta que se le da. Ocurre en el razonamiento backward.

VENTAJAS DE LOS SBR.

- Modularidad: Cada regla es una unidad de conocimiento que puede ser añadida, modificada o removida independientemente de las otras reglas existentes. Esto da flexibilidad en el desarrollo de la Base de Conocimientos (BC).
- Uniformidad: Todo el conocimiento del sistema se expresa en el mismo formato.
- Naturalidad: Las reglas son un formato natural para expresar conocimiento en algunos dominios. Los expertos lógicamente piensan en los problemas y sus soluciones usando las situaciones existentes para indicar las conclusiones deseadas.
- Son fáciles de programar y manejar ya que pueden codificarse las reglas en cualquier orden.

DESVENTAJAS DE LOS SBR.

Los problemas de los SBR caen en tres categorías:

- Encadenamiento infinito.
- Adición de nuevo conocimiento que puede resultar contradictorio.

- Modificación de reglas existentes.

1.2.1.2 Basados en casos (SBC).

Los SBC son una de las tecnologías actuales para construir Sistemas Basados en el Conocimiento. En ellos, los nuevos problemas se resuelven considerando la solución dada a problemas similares resueltos en el pasado. La arquitectura básica de un SBC consiste de una base de casos, un procedimiento para buscar casos similares y un procedimiento de adaptación para ajustar las soluciones de los problemas similares a los requerimientos del nuevo problema.

Dentro de las Técnicas de IA, los SBC son capaces de utilizar el conocimiento específico adquirido en situaciones previas y utilizarlo en la situación presente. El problema nuevo se resuelve buscando en su memoria un caso similar resuelto en el pasado. Además incrementa su conocimiento almacenando el nuevo caso para ser usado en situaciones futuras.[\[5\]](#)

Razonamiento Basado en Casos significa razonar sobre la base de experiencias o "casos" previos. Los SBC es una alternativa entre otras metodologías para construir sistemas basados en conocimiento. Al razonar basado en casos, el solucionador de problemas recuerda situaciones previas similares a la actual y las usa para ayudar a resolver el nuevo problema.

La idea básica de los SBC es recuperar, adaptar y validar las soluciones encontradas en experiencias previas en un intento de relacionarlas con un problema actual. Las experiencias previas están representadas como una biblioteca de casos que reside en memoria. Cuando se enfrenta con un nuevo problema, los SBC recuperan un caso similar, y la solución del caso se adapta al nuevo problema en un intento para resolverlo.

Estos SBC son posible aplicarlos cuando el conocimiento disponible es escaso y los modelos débiles. Los problemas no son completamente comprendidos, existen muchas excepciones a las reglas o bien en caso de ser posible formularlas, su uso resulta caro ya que constituyen una base demasiado grande y el encadenamiento promedio es demasiado largo.

COMPONENTES DE UN SBC.

Las componentes fundamentales de un SBC son: la base de casos, el módulo de recuperación y el módulo de adaptación.

LA ESTRUCTURA DE ESTE TIPO DE SISTEMAS SE DIVIDE EN CUATRO PASOS.

1. Recuperar los casos más parecidos donde un nuevo problema se aparea con casos similares guardados en la base de casos.
2. Reutiliza la solución propuesta en los casos para tratar de resolver el problema.
3. Revisar la solución propuesta.
4. Almacenar la nueva solución como parte de un nuevo caso.

LOS SISTEMAS SBC SE CLASIFICAN EN UNO DE LOS TRES NIVELES SIGUIENTES, SEGÚN EL CONJUNTO DE FACILIDADES QUE PROPORCIONEN.

- Sistemas consejeros que tan sólo recuperan casos: Sin las facilidades de adaptación, evaluación y reparación. El usuario describe una situación y el sistema devuelve casos previos relevantes de los que el usuario extraerá sus propias conclusiones.
- Sistemas con recuperación y adaptación: Sin evaluación ni reparación. El usuario describe una situación, y el sistema encuentra y adapta casos previos similares.
- Sistemas con recuperación, adaptación y reparación: El usuario describe una situación y el sistema encuentra y adapta casos previos relevantes. Si la solución propuesta falla, el sistema la repara y también modifica el mecanismo de recuperación para evitar el mismo error en el futuro.

VENTAJAS DE LOS SISTEMAS BASADOS EN CASOS.

- Adquisición de conocimiento: La unidad básica del conocimiento es el caso. Los seres humanos por lo general articulan su conocimiento mediante ejemplos de problemas y soluciones anteriores (casos), más que por medio de reglas específicas y abstractas.
- Permite proponer soluciones a problemas rápidamente: Esto lo logra ya que las respuestas no se derivan a partir de cero, sino de casos resueltos previamente.
- Aprendizaje: Si la misma situación se presenta repetidamente, no se tiene que construir o generar la misma solución a partir de cero.
- Propone soluciones en dominios no entendidos completamente por el sistema.
- Ofrece un medio de evaluación de soluciones cuando no se cuenta con un método algorítmico.
- Se centra en las características o partes más importantes del problema.

DESVENTAJAS DE LOS SISTEMAS BASADOS EN CASOS.

- Confía ciegamente en los casos previos almacenados en su memoria para intentar proponer su solución.
- Puede ser que no recupere el caso más apropiado para la solución del nuevo caso.

1.2.1.3 Basados en redes bayesianas.

Una red bayesiana puede definirse como un grafo orientado y sin ciclos en el que los nudos representan variables aleatorias que pueden ser continuas o discretas y los arcos que los encadenan expresan las relaciones entre dichas variables. En la red o grafo, cada variable aleatoria recibe la asignación de su probabilidad de comportamiento, así como las probabilidades condicionales correspondientes.

Las redes bayesianas o probabilísticas se fundamentan en la teoría de la probabilidad y combinan la potencia del teorema de Bayes con la expresividad semántica de los grafos dirigidos; las mismas

permiten representar un modelo causal por medio de una representación gráfica de las independencias / dependencias entre las variables que forman parte del dominio de aplicación.

FORMALMENTE SE DEFINE COMO RED BAYESIANA UNA TRIPLETA (N,D,P) DONDE:

- N es un conjunto de variables del dominio.
- D es una DAG (Grafo acíclico dirigido) cuyos nodos están etiquetados con los elementos de N y los arcos dirigidos indican relación de influencia y en algunos casos relación causal.
- P es una distribución joint sobre N.
- D reúne la información de que toda variable i es independiente de sus no descendientes dados sus padres (Padres (i)).

SE PUEDE INTERPRETAR A UNA RED BAYESIANA DE DOS FORMAS:

1. Distribución de probabilidad: Representa la distribución de la probabilidad conjunta de las variables representadas en la red.
2. Base de reglas: Cada arco representa un conjunto de reglas que asocian a las variables involucradas. Dichas reglas están cuantificadas por las probabilidades respectivas.

REPRESENTACIÓN DEL CONOCIMIENTO.

Una red bayesiana representa relaciones causales en el dominio del conocimiento a través de una estructura gráfica y las tablas de probabilidad condicional entre los nodos, por lo tanto el conocimiento que representa la red está compuesto por los siguientes elementos:

- Un conjunto de nodos $\{ X_i \}$ que representan cada una de las variables del modelo. Cada una de ellas tiene un conjunto exhaustivo de estados $\{ x_i \}$ mutuamente excluyentes.
- Un conjunto de enlaces o arcos (X_i, X_j) entre aquellos nodos que tienen una relación causal. De esta manera todas las relaciones están explícitamente representadas en el grafo.

- Una tabla de probabilidad condicional asociada a cada nodo X i indicando la probabilidad de sus estados para cada combinación de los estados de sus padres. Si un nodo no tiene padres se indican sus probabilidades a priori.

COMPONENTES PRINCIPALES DE UNA RED BAYESIANA: CUALITATIVO Y CUANTITATIVO.

- En el campo cualitativo tenemos un grafo acíclico dirigido en el que cada nodo corresponde a un atributo (variable), y arcos dirigidos implicando que toda variable es condicionalmente independiente de todos sus no descendientes en la red siempre que se conozcan los valores de sus inmediatos predecesores (padres). Una variable Z es descendiente de otra variable Y , si en el grafo existe un camino dirigido desde Y a Z , por ejemplo en el grafo expuesto I es descendiente de A .
- En el campo cuantitativo cada nodo tiene asociada la distribución de probabilidad de esa variable teniendo en cuenta sus padres en el grafo. Por ejemplo para el nodo I , tendremos $P(I | B, H)$.

SEPARACIÓN DIRECCIONAL.

Dado un grafo dirigido acíclico conexo y una distribución de probabilidad sobre sus variables se dice que hay una separación direccional si, dado un nodo X , el conjunto de sus padres separa condicionalmente ese nodo de todo otro subconjunto Y en que no haya descendientes de X .

$$P(x | pa(x), y) = P(x | pa(x))$$

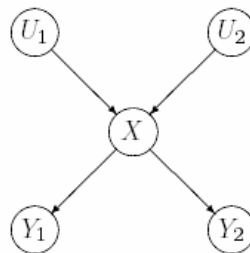


Fig. 2 Separación Direccional

Una vez conocido el valor de x podemos calcular la probabilidad de y_1 sin que influyeran los valores de las demás variables, es decir, el conjunto $pa(y_1)=\{x\}$, separa condicionalmente a y_1 de todas las demás variables de la red. [6]

PROCESO DE INFERENCIA.

La utilidad para la IA radica en que se puede realizar inferencia sobre la red obteniendo los valores más probables de las causas en función de los efectos o viceversa. Su base teórica principal es la probabilidad condicionada, los teoremas de Bayes y el teorema de la probabilidad total.

En una red bayesiana se pueden realizar dos tipos de inferencia:

- Inferencia causal o descendente: Conocemos la distribución de probabilidad de algunos de los nodos causa y queremos inferir la distribución de alguno de los nodos efecto. Dado $P(C)$, ¿Cuál es el valor de $P(E|C)$? Esta inferencia se representa mediante la letra griega π .
- Inferencia de diagnóstico o ascendente: Conocemos la distribución de probabilidad de alguno de los nodos efecto y queremos inferir la distribución de algún nodo causa. Dado $P(E)$ ¿Cuál es el valor de $P(C|E)$? Esta inferencia se representa mediante la letra λ .

VENTAJAS DE LAS REDES BAYESIANAS.

Las redes bayesianas presentan toda la información en un único formato (probabilístico y gráfico) lo que hace sencillas las interpretaciones, permiten retractarse de conclusiones obtenidas con anterioridad que ya no nos son razonables a la luz de las nuevas evidencias, nos proporcionan una visión general del problema, generan un conjunto de alternativas ordenadas y facilita la explicación de las conclusiones [7]. Por otro lado, cuando construimos una red bayesiana a partir del conocimiento de un experto para usarla en la orientación ante la toma de decisiones, la asignación de posibilidades es sencilla [8]. Además, las redes bayesianas permiten trabajar con conceptos de la teoría de la decisión como valor o valor esperado frente a problemas de decisión.

En el contexto de la inferencia, las redes bayesianas permiten realizar inferencias bidireccionales, esto es, desde los efectos a las causas y desde las causas a los efectos. Y lo que es más interesante, permiten llevar a cabo inferencias abductivas, o sea, encontrar la mejor explicación para un conjunto de datos. Sin embargo, la ventaja más importante de una red bayesiana en el ámbito de la inferencia estadística se deriva de su habilidad para realizar computaciones locales.

Esta propiedad permite que la actualización de la probabilidades se lleve a cabo eficientemente sin tener que calcular todas las posibles combinaciones entre todos los niveles de las variables (que supondría un incremento exponencial de los cálculos de medida que aumenta el número de variables del modelo o el número de estados por variable) cuando se incorpora determinado conocimiento a la estructura gráfica. Gracias a esta propiedad derivada de los principios de dependencia e independencia condicional, las redes bayesianas son herramientas especialmente indicadas para tareas que requieren una actualización rápida y continua en los procesos de control, por ejemplo, industriales.

DESVENTAJAS DE LAS REDES BAYESIANAS.

Una de las grandes desventajas de su empleo es la gran cantidad de probabilidades numéricas que se necesita conocer para su utilización. Además, el empleo de estas redes implica un alto costo computacional y así mismo por la complejidad de la implementación de los algoritmos de inferencia se requiere de gran espacio en memoria. También para desarrollar una red de este tipo es necesaria la realización de estudios estadísticos.

1.2.2 Redes Neuronales Artificiales (RNA).

Una RNA, según Freeman y Skapura [\[9\]](#), es un sistema de procesadores paralelos conectados entre sí en forma de grafo dirigido. Esquemáticamente cada elemento de procesamiento o neuronas de la red se representa como un nodo. Estas conexiones establecen una estructura jerárquica que tratando de emular la fisiología del cerebro busca nuevos modelos de procesamiento para solucionar problemas concretos del mundo real. Lo importante en el desarrollo de la técnica de las RNA es su útil comportamiento al aprender, reconocer y aplicar relaciones entre objetos y tramas de objetos propios

del mundo real. En este sentido, se utilizan las RNA como una herramienta que podrá utilizarse para resolver problemas difíciles.

La posibilidad de resolver problemas difíciles es dada gracias a los principios de las redes neuronales, los cinco más importantes son citados por Hilera y Martínez. [\[10\]](#)

PRINCIPIOS DE LAS REDES NEURONALES:

1. Aprendizaje adaptativo: Pueden comportarse en función de un entrenamiento con una serie de ejemplos ilustrativos. De esta forma, no es necesario elaborar un modelo a priori, ni establecer funciones probabilísticas. Una RNA es adaptativa porque puede modificarse constantemente con el fin de adaptarse a nuevas condiciones de trabajo.
2. Auto organización: Mientras que el aprendizaje es un proceso donde se modifica la información interna de la RNA, la auto organización consiste en la modificación de la red completa con el fin de llevar a cabo un objetivo específico. Auto organización significa generalización, de esta forma una red puede responder a datos o situaciones que no ha experimentado antes, pero que puede inferir sobre la base de su entrenamiento. Esta característica es muy útil sobre todo cuando la información de entrada es poco clara o se encuentra incompleta.
3. Tolerancia a fallos: En la computación tradicional la pérdida de un fragmento pequeño de información puede acarrear comúnmente la inutilización del sistema. Las RNA poseen una alta capacidad de tolerancia a fallos. Se entiende por ello que las redes pueden reconocer patrones de información con ruido, distorsión o incompletos, pero que, además, pueden seguir trabajando aunque se destruya parte de la red (con cierta degradación). La explicación de este fenómeno se encuentra en que mientras la computación tradicional almacena la información en espacios únicos, localizados y direccionables, las redes neuronales lo hacen de forma distribuida y con un alto grado de redundancia.
4. Operación en tiempo real: Las RNA, de todos los métodos existentes, son las más indicadas para el reconocimiento de patrones en tiempo real, debido a que trabajan en paralelo actualizando todas sus instancias simultáneamente. Es importante destacar que esta

característica solo se aprecia cuando se implementan redes con hardware especialmente diseñados para el procesamiento paralelo.

5. Fácil inserción en la tecnología existente: Es relativamente sencillo obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilita la integración modular en los sistemas existentes.

VENTAJAS QUE POSEEN LAS REDES NEURONALES.

- Tienen una gran eficacia de clasificación y son más robustas que la mayor parte de las técnicas de tipo estadístico.
- No depende, a diferencia de otros enfoques estadísticos, de hipótesis sobre la distribución de los valores o la independencia de los atributos.

DESVENTAJAS DE LAS RNA.

- Los algoritmos de entrenamiento suelen ser bastante lentos, si se comparan con el proceso de construcción de árboles y reglas de los algoritmos de aprendizaje inductivo.
- El conocimiento que se obtiene (pesos y umbrales) es ininteligible para los seres humanos.
- En ocasiones, es difícil determinar la configuración idónea de la red para la resolución de un problema concreto.

1.3 LOS SISTEMAS SCADA.

Se conoce como el nombre de SCADA (Supervisory Control And Data Acquisition o Control con Supervisión y Adquisición de Datos) a cualquier software que permita el acceso a datos remotos de un proceso y permita, utilizando las herramientas de comunicación necesarias en cada caso, el control del mismo. [\[11\]](#)

1.3.1 Ventajas del SCADA.

- La modularidad de los autómatas permite adaptarlas a las necesidades actuales y ampliarlos posteriormente si es necesario.
- Cualquier tipo de sensores y actuadores puede integrarse en un programa de Control Lógico Programable (PCL) mediante las múltiples tarjetas de adquisición disponibles (tensión, corriente, sondas de temperatura).
- Gracias a las herramientas de diagnósticos se consigue una localización más rápida de los errores. Esto permite minimizar los periodos de paro en las instalaciones y repercute en la reducción de costes de mantenimiento.
- Los programas de visualización pueden presentar todo tipo de ayuda al usuario, desde la aparición de una alarma hasta la localización de las causas o la parte del esquema eléctrico implicada en la misma.
- La integración de sistemas es rápida gracias a los sistemas de comunicación estandarizados.
- Aumento de la calidad del producto mediante las herramientas de diagnósticos. El operador es notificado en el momento en que se detecta una incidencia.
- La distribución de recursos y control sobre la red permite una mejor coordinación entre las estaciones remotas en caso de fallos en una de ellas. [\[12\]](#)

1.3.2 Subsistemas de SCADA.

El SCADA está constituido por los subsistemas observados en la **Fig 3**. De los cuales se tratarán solo los que estén vinculados de una forma u otra al tratamiento de las alarmas en el sistema. Estos serían el sub-sistemas de adquisición, el subsistema middleware y el subsistema de visualización.

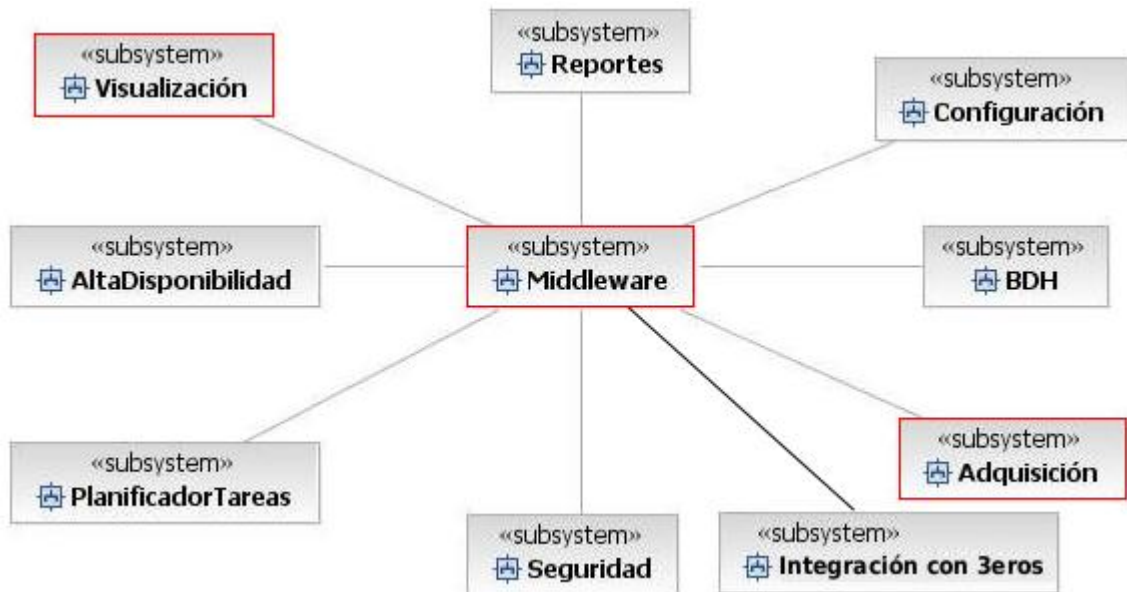


Fig. 3 Arquitectura General del SCADA

1.3.2.1 Subsistema de Adquisición.

La adquisición está formada por los siguientes componentes:

- Manejadores de Protocolos.
- Módulo de Recolección.
- Base de Datos de Tiempo Real, (BDTR).

1- MANEJADORES DE PROTOCOLOS (DRIVERS).

En la cadena de tratamiento de la información en un SCADA el primer eslabón es la adquisición de los datos. Estos provienen de disímiles equipos que pueden ser autómatas, PLC, reguladores autónomos, sensores inteligentes, controladores, etc.

La función principal de los drivers es enlazar el SCADA con los diferentes dispositivos del campo. La interacción entre los manejadores y los dispositivos se efectúa mediante el envío y recepción de

mensajes a través de un medio físico determinado. Los manejadores se encargan de la adquisición de los datos traduciendo los protocolos de campo a un protocolo genérico. Se permite la programación de estos en módulos independientes al sistema para evitar la recompilación del sistema ante incorporaciones de nuevos protocolos.

2- MÓDULO DE RECOLECCIÓN.

Las tareas de recolección de la información de campo deben ser planificadas en función de los tipos de protocolos y redes de comunicación. La adquisición se realiza a través de los manejadores anteriormente descritos ya sean por mecanismo de encuesta-respuesta o por mecanismos de escucha.

El recolector es responsable de construir las listas de los puntos asociados a dispositivos de campo, a consultar agrupados según la frecuencia de recolección y entregar dichas listas a los manejadores para que estos construyan los bloques de encuesta.

La información adquirida por los manejadores es entregada a la BDTR, a través de una conexión etiquetada como *BDTR Output Connector*.

Punto importante a tener en cuenta es que la planificación de las tareas se realiza en función de la configuración previamente realizada en el entorno de edición del sistema por los mantenedores del proyecto, lo cual se realiza a través de un conector etiquetado como *Configurator*.

3- BASE DE DATOS DE TIEMPO REAL.

La BDTR es el módulo responsable del llevar a cabo este procedimiento mediante la ejecución de tareas especializadas que difieren según el tipo de señal que recibe.

El Módulo BDTR está encargado de manejar todo lo referente a la recepción, procesamiento y distribución de los datos proveniente de campo en tiempo real y de esta forma hace posible la ejecución de toda la lógica del sistema, permitiendo la ejecución de acciones como:

- Adquisición, en tiempo real, de datos del nivel de recolección.

- Conversión de unidades y linealización de datos recolectados.
- Procesamiento de variables calculadas.
- Detección y manejo de Alarmas.
- Control de Calidad de los datos recolectados.
- Publicación a los clientes de la actualización de los puntos, sucesión de alarmas y eventos.
- Propagación, a los niveles inferiores, de los comandos enviados por los operadores o sistemas automatizados de control de procesos.

1.3.2.2 Subsistema Middleware.

El Middleware es la capa de software, que se encarga de la comunicación entre los diferentes módulos que forman parte del sistema.

Este módulo tiene como finalidad proporcionar la capa de comunicación de alto nivel, tanto sincrónica, como asincrónica, para la comunicación de todos los módulos que conforman el SCADA.

El Middleware del SCADA proporciona mecanismos asincrónicos para el envío de puntos, alarmas, comandos, lotes de puntos, lotes de alarmas, eventos y bitácoras, así mismo, también brinda una interfaz que permite a los módulos del sistema, ejecutar funcionalidades presentes en otros módulos de manera sincrónica, mediante llamadas a procedimientos remotos en el marco de una arquitectura cliente/servidor.

1.3.2.3 Subsistema de Visualización (HMI).

El módulo de HMI en el SCADA se encarga de representar, en un ordenador, los procesos que ocurren en el campo en tiempo real, muestra los componentes implicados, los sensores, las estaciones remotas, y el sistema de comunicación dándole al operador total control. Éste módulo es el que permite al operador estar en contacto directo con el sistema y realizar la supervisión y el control del proceso en general. Para un mejor aprovechamiento y explotación de las facilidades que brinda la interfaz en un

SCADA es necesario el conocimiento del funcionamiento y manejo de la misma.

El subsistema de visualización del SCADA se divide en tres aplicaciones fundamentales:

1. Ambiente de Ejecución (AE).
2. Ambiente de Configuración (AC).
3. Reportes.

FUNCIONALIDADES GENERALES.

- Permitir el monitoreo de la información del proceso, en tiempo real, utilizando para ello interfaces gráficas que modelen o simulen de forma simple y accesible el proceso.
- Permitir a los operadores el envío comandos para hacer modificaciones de los parámetros de control o configuración del proceso.
- Visualizar los datos históricos y en tiempo real en forma de gráficos de tendencia.
- Visualización, filtrado y manejo de alarmas y eventos.
- Configuración de los recursos del sistema, módulos, puntos, alarmas, recetas, etc.[\[13\]](#)

1.3.3 Las Alarmas en el Sistema.

SUMARIO DE ALARMAS.

El sumario de alarmas es una interfaz gráfica que concentra las condiciones de procesos críticas, medias y baja presentes en el sistema y cuyo objetivo primordial es guiar al operador a la detección del origen de la falla y supervisar la ejecución de las medidas de corrección automatizada o manual.

PRINCIPIO DE FUNCIONAMIENTO DEL SUMARIO.

Los tipos de alarmas incluyen:

Momentánea: Alarma que desaparece del sumario, al retornar la variable al estado normal independientemente de si ha sido reconocida o no.

Mantenida: Alarma que desaparece del sumario, solo cuando la variable retorna al estado normal y la misma ha sido reconocida.

NIVEL DE SEVERIDAD.

La jerarquía de la alarma se define por su nivel de severidad, lo cual determina su tratamiento, según lo siguiente:

Severidad 1 (Alarmas Críticas).

Las alarmas también conocidas como “Alarmas Críticas”, son aquellas que requieren acción inmediata del operador:

1. Después que la acción correctiva automática ha sido activada.
2. Cuando la acción correctiva automática no está disponible.

En ambos casos el retardo o el fracaso de la acción correctiva pueden causar heridas al personal, producir daños ambientales o afectar las instalaciones, dañar o causar otras pérdidas sustanciales.

Severidad 2 (Media).

El objetivo de estas alarmas es advertir al operador que es necesario tomar medidas a fin de eliminar la desviación que activó la alarma y evitar la ejecución de alguna acción correctiva automática o manual. Un ejemplo típico de esto son las “pre alarmas”.

Severidad 3 (Advertencia de Alarmas).

Usado para objetivos de información relacionados con desviaciones de proceso, equipo de proceso estado anormal, estado de cierre no incluido en prioridad 1 ó 2, alarmas de sistema, etc.

TRATAMIENTO DE LA ALARMA SEGÚN TIPO.

Alarmas de tipo Momentánea:

Este tipo de alarmas, suceden en las entidades en las cuales han sido configuradas o son establecidas previamente por diseño en el proyecto, y se mantienen en sumario mientras exista la condición anormal, desapareciendo del mismo al restablecerse la condición normal de la entidad. El reconocimiento o no de este tipo de alarmas no tiene influencia directa en su permanencia dentro del sumario.

Alarmas clasificadas dentro de este tipo:

- Alarmas de Falla de Comunicación en dispositivos o sub-canales.
- Alarmas de Falla de Instrumento.
- Alarma de Falla de no Variación en el Tiempo.

Alarmas de tipo Mantenido:

Este tipo de alarmas, suceden en las entidades en las cuales han sido configuradas, o son establecidas previamente por diseño en el proyecto, y su permanencia dentro del sumario está determinada por la existencia de dos condiciones: la persistencia del estado anormal y el reconocimiento de la misma. La alarma sólo puede borrarse del sumario al restablecerse la condición normal y ser reconocida por el operador del sistema.

Alarmas clasificadas dentro de este tipo.

- Alarmas de Falla de Ejecución de Comando.
- Alarmas de Cambio de Estado no Comandado.

- Alarmas de Tasa de Cambio.
- Alarmas de Cambio de Estado.
- Alarmas de Nivel.
- Alarma de Desviación.

MODOS DE PRESENTACIÓN.

El sumario es representado como una lista de alarmas en modo contraído o expandido.

Sumario en Modo de lista contraída.

Muestra las cinco (5) últimas alarmas independientemente del orden presentado en el sistema y ofrece barra de desplazamiento para visualizar las alarmas en su orden de aparición.

Sumario en Modo de lista expandida.

Representa las alarmas ordenadas de acuerdo a la relación severidad, prioridad, reconocimiento.

La lógica de representación en sumario expandido. [\[14\]](#)

CARACTERISTICAS DE LAS ALARMAS.

1. Alarmas de Nivel.

Estas alarmas son las encargadas de verificar que los valores de los puntos analógicos se encuentren dentro de un rango de operación preestablecido.

Tipos de alarmas de nivel.

- Alarmas de nivel Alto - Alto (High - High).
- Alarmas de nivel Alto (High).
- Alarmas de nivel Bajo (Low).
- Alarmas de nivel Bajo - Bajo (Low - Low).

2. Alarma de Cambio de Estado No Comandado.

Esta alarma tiene la función de identificar cuando ocurre un cambio en un punto que no ha sido comandado por el sistema. Para realizar la activación de la alarma se compara el valor recolectado con el último valor comandado por el sistema, en caso de ser diferente la alarma se activa.

3. Alarma de Falla de Ejecución de Comando.

Esta alarma tiene la función de verificar si un comando enviado fue ejecutado con éxito o no. Cuando un comando llega a la BDTR, se guarda el estado actual que tiene el punto antes de ejecutar el nuevo comando **Previous Value**; posteriormente al llegar la respuesta de dicho comando, en caso de ser un punto digital se construye el **Control Value** aplicando el comando al **Previous Value**, de esta manera se obtiene el estado actual que debería tener el punto, este proceso se realiza de esta manera para poder tratar a los comandos digitales simples, que solo cambian un segmento de la variable. En el caso del punto Analógico la asignación del **Control Value** es directa con el valor del comando (tomando en cuenta que se debe realizar una linealización, porque el **Control Value** siempre debe estar en unidades de ingeniería.

En la tarea de respuesta del comando, si el estado del mismo es fallido, se realiza la petición de procesamiento inmediato de la Alarma de Falla de Ejecución de Comando, en cambio, si el comando fue ejecutado con éxito, se realiza una petición temporizada para la verificación de la alarma, utilizando el tiempo de ejecución de comando **transferTime** como tiempo de espera para el procesamiento de la tarea de alarma.

4. Alarmas de Tasa de Cambio.

La Alarma de Tasa de Cambio monitorea que el cambio, en un tiempo determinado, no supere el valor asignado:

$$\text{Tasa de Cambio} = \text{abs} ((\text{variable.PVt-1} - \text{variable.PVt0}) / (\text{variable.PV.time-1} - \text{variable.PV.time0}))$$

Donde los tiempos **variable.PV.time** están expresados en segundos y los datos **variable.PV** esta expresado en las unidades de ingeniería de la variable.

5. Alarmas de Desviación.

Las señales involucradas en el control del proceso, en muchas ocasiones operacionales necesitan mantenerse dentro de rango de operación que puede depender de otras variables o funciones lógicas. La Alarma de Desviación se presenta cuando la señal se compara con el resultado de la expresión de referencia pudiendo ser esta Punto, función aritmética/lógica o un número constante y esta supera el umbral de desviación configurado para la misma según la siguiente ecuación:

$$\text{Desviación} = \text{Valor actual} - \text{Valor Expresión de Referencia.}$$

Si Desviación > Umbral, la Alarma se activa.

6. Alarmas de Falla de Instrumento.

Esta alarma se activa cuando existe un fallo en un sensor o el sensor está fuera de calibración de acuerdo a diagnósticos internos, puede ocurrir también que el valor retornado por un dispositivo está fuera de los límites definidos para el parámetro, así mismo si un driver está fuera de los límites se activa también.

Cuando se trata de calidad de dato “Buena” el sistema verifica el valor recibido contra los valores de los límites inferior y superior para la entrada en unidad de ingeniería en la configuración del punto. Si el valor recibido se encuentra fuera de este rango se tratará como una falla de instrumento

7. Alarmas de Falla de Comunicación.

Las fallas de comunicación para dispositivos y sub-canales, están asociadas al COMMSTATE, si la alarma se encuentra configurada para el dispositivo el sistema verifica el estado de comunicaciones para el dispositivo o sub-canal, luego en caso de que la trama de calidad indique FALLA, incrementa un contador y verifica el número máximo de reintentos antes de decretar la alarma, si el contador excede la cantidad de reintentos el sistema genera la alarma con la última trama de calidad obtenida del COMMSTATE.

En caso de obtener una trama buena antes de cumplirse el máximo número de reintentos, el contador de tramas malas vuelve a cero.

Las alarmas asociadas a Sub-Canal y Dispositivos se generan de forma independiente para cada entidad, es decir se obtiene el COMMSTATE para cada entidad independiente y se realiza el tratamiento de la alarma.

Si existe una falla a nivel de sub-canal y la misma afecta a los dispositivos asociados, las alarmas se generarán, según lo que indique el COMMSTATE de cada entidad.

8. Alarmas de Falla de no Variación en el Tiempo.

Cuando se encuentra configurada, el sistema generará una alarma cuando un valor analógico no ha cambiado después de un número configurable de barridos de lectura. Esta alarma indica que existe la posibilidad que el sensor analógico esté “muerto”. Cuando en alguna de las subsecuentes lecturas hay un cambio en el valor, el sistema desactivará la alarma.

9. Alarmas de Cambio de Estado.

Se compara el valor del BIT o los BITS asociados con los configurados para el punto. Si es igual al BIT o los BITS asociados a la alarma entonces se genera la alarma.

Estados lógicos para producir una alarma:

- Valor Lógico 0 (Falso).
- Valor Lógico 1 (Verdadero).
- Cambio de estado a 1 (0 à 1).
- Cambio de estado a 0 (1 à 0).[\[15\]](#)

1.4 SELECCIÓN DE LA TÉCNICA A UTILIZAR.

Después de realizado el presente estudio se ha seleccionado los SBR como técnica a utilizar para la toma de decisiones en el futuro módulo de IA para el tratamiento de las alarmas en el sistema SCADA UX.

Estos sistemas tienen una mayor flexibilidad en el desarrollo de la base de conocimientos, ya que esta puede ser modificada fácilmente, gracias a que todo el conocimiento del sistema se expresa en un mismo formato. Para resolver el problema de la gestión de las alarmas del SCADA no es necesario utilizar ningún tipo de método de entrenamiento, ya que utilizando las reglas fijas del SBR se puede disponer de la información necesaria para obtener un resultado satisfactorio.

Los SBC no son propicios para el módulo de IA ya que estos trabajan con la recuperación y adaptación de casos previos similares al actual, esto trae consigo un gran número de excepciones a las reglas y el encadenamiento promedio es muy largo. Además siendo el SCADA un sistema configurable existiría gran variedad de casos, por lo que resultaría más complicada la inferencia.

Los sistemas Basados en Redes Bayesianas no son adecuados para el sistema SCADA UX ya que al ser este configurable, no se conoce con exactitud la cantidad de datos con la que se pondrá el sistema en funcionamiento. Los sistemas Basados en Redes Bayesianas trabajan sobre todo con la probabilidad, pero el sistema puede o no tener los datos necesarios para que el sistema probabilístico funcione, ya que se necesita para su funcionamiento un gran número de probabilidades numéricas, las cuales son más difíciles de obtener con sistemas que trabajan con valores en tiempo real.

Las RNA son muy utilizadas en la toma de decisiones, pero para que estas trabajen es necesario aplicarles algoritmos de entrenamiento, principalmente un proceso de prueba y error, el cual definirá la configuración que tendrá la red neuronal para un problema en específico. Para realizar este proceso de adaptación y configuración es necesario un conjunto de datos de entrenamiento, el cual no se tiene al iniciar el sistema SCADA UX, pues el mismo se configura en dependencia de la industria en la cual se vaya a instalar, por lo que se vería afectada en gran medida la utilización de las RNA para la gestión de las alarmas de dicho sistema.

1.5 HERRAMIENTAS Y METODOLOGÍAS UTILIZADAS.

1.5.1 RUP como metodología de desarrollo.

El Proceso Unificado de Rational (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis y diseño, implementación y documentación de sistemas orientados a objetos, principalmente de grandes proyectos como el caso del SCADA. La metodología RUP, fue desarrollada en 1998 por Grady Booch, Ivar Jacobson y James Rumbaugh, reconocidos metodólogos en la industria de la tecnología y sistemas de información. RUP se caracteriza por ser: dirigido por Casos de Uso, centrado en la arquitectura, iterativo e incremental. Divide el proceso de desarrollo en ciclos o fases, teniendo un producto final al culminarse cada una de las interacciones por fases [\[16\]](#). Con el uso de una metodología se pretende reducir costos y retrasos de proyectos, así como mejorar la calidad del software.

1.5.2 Visual Paradigm.

Visual Paradigm es una herramienta que utiliza UML como lenguaje de modelado, resulta intuitiva y fácil de utilizar, soporta el ciclo de vida completo del desarrollo de software. Brinda la posibilidad de construir la aplicación de forma rápida, con mayor calidad y menor coste. Permite dibujar todos los tipos de diagramas de clases, utilizar código inverso, además de generar código desde diagramas y generar documentación. Esta herramienta proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML. Cuenta con un editor de detalles de casos de usos y con un entorno integrado para la especificación de los detalles de cada caso de uso. Permite la importación y exportación de ficheros XML y corre en plataformas como Windows, Linux y Mac OS X.

1.5.3 UML como lenguaje de modelado.

UML (Unified Modeling Language) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica

esencial de lo que estos diagramas y símbolos significan^[17]. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. UML es una consolidación de muchas de las notaciones y conceptos más usados orientados a objetos. La última versión, UML 2.0, proporciona a los analistas, arquitectos y desarrolladores; herramientas más potentes que les posibilita aprovechar mejor los modelos y generar así una mayor cantidad de código reduciendo en gran medida el ciclo de desarrollo de sus aplicaciones.

Características principales.

- Divide cada proyecto en un número de diagramas que representan las distintas vistas del proyecto y juntos representan la arquitectura del mismo.
- Permite describir un sistema en diferentes niveles de abstracción, simplificando la complejidad sin perder información, para que los usuarios y desarrolladores comprendan las características de la aplicación.
- Se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del desarrollo de una aplicación, sin embargo no pretende definir un modelo de desarrollo sino únicamente un lenguaje de modelado.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.

En este capítulo se aborda el proceso de diseño, el cual va a ser representado mediante artefactos ingenieriles, tales como los diagramas de clases de diseño, de secuencia, y los de caso de usos del sistema, así como también quedarán reflejadas las reglas del negocio, los requisitos funcionales y los no funcionales, todo esto con vista a darle solución al demo de IA para la gestión de las alarmas del SCADA UX.

2.1 REGLAS DEL NEGOCIO

Las reglas de la base de conocimientos son las soluciones que se le dan a las alarmas emitidas por el sistema, y se basarán en el punto que generó la alarma y el tipo de alarma.

Estas soluciones serán definidas por los operadores expertos de cada sistema, de acuerdo a la industria en la que se instale y serán configurables de acuerdo con las nuevas especificaciones o cambios que puedan ocurrir en el entorno industrial.

Para definir el orden en que se tratarán las alarmas se tendrán en cuenta las variables prioridad, severidad, estado, de la siguiente forma:

- El aspecto más importante en las alarmas es su severidad por lo que este determinará en primer lugar el orden, las alarmas de severidad crítica serán las primeras en atender.
- Se ordenan de la forma: todas las Alarmas Activas no Reconocidas, ordenadas por su prioridad. Alarmas con la misma prioridad se comportan en modo LIFO dentro de su grupo.
- Luego se ubican las alarmas con severidad media, posterior al grupo de alarmas con severidad crítica igualmente en el orden Activas no Reconocidas ordenadas por su prioridad. Alarmas con la misma prioridad se comportan en modo LIFO dentro de su grupo.

- Posteriormente se ubican las alarmas de severidad baja, después del grupo de alarmas con severidad media no reconocida ordenadas por su prioridad. Alarmas con la misma prioridad se comportan en modo LIFO dentro de su grupo.
- Luego se ubican las alarmas Activas Reconocidas, ordenadas por su severidad, posterior al grupo de alarmas con Severidad baja no reconocidas. Alarmas con la misma prioridad se comportan en modo LIFO dentro de su grupo.
- De esa misma forma se ubicarán las Alarmas No Activas no Reconocidas.
- Las Alarmas no Activas no Reconocidas desaparecen.

2.2 CAPTURA DE REQUISITOS.

2.2.1 Requisitos funcionales.

Los requisitos funcionales son características requeridas del sistema que expresa una capacidad de acción, funcionalidad o condición que el sistema debe cumplir. Estos requisitos siempre se van a mantener invariables sin importar con que propiedades o cualidades se relacionen.

Requisitos funcionales del módulo de IA para el tratamiento de alarmas.

RF1. Obtener Alarmas.

RF1.1 Recibir alarmas.

RF2. Cargar reglas.

RF2.1 Acceder a la base de conocimiento.

RF2.2 Asociar respuestas (reglas de la base de conocimientos) a cada alarma.

RF3 Obtener respuesta.

RF3.1 Obtener lista de posibles soluciones organizadas.

RF3.2 Enviar notificaciones de soluciones al operador.

RF4. Gestionar Reglas.

RF4.1. Crear Regla.

RF4.2. Modificar Regla.

RF4.3. Eliminar Regla.

2.2.2 Requisitos no funcionales.

Los requisitos no funcionales son diferentes propiedades que el producto debe tener. Estas propiedades hacen a su vez que el producto sea atractivo, usable, rápido y confiable teniendo estas como características principales.

1. USABILIDAD

- El sistema podrá ser usado por aquellos usuarios que posean conocimientos básicos de los procesos.
- El sistema permitirá tener acceso a los datos.

2. CONFIABILIDAD

- El sistema guarda automáticamente la información existente si se produce algún tipo de error o falla que provoque el cierre del sistema.
- Si existe algún error a la hora de ingresar datos el sistema, sistema se mantiene con los datos guardados antes de dicha operación.

3. EFICIENCIA

- El sistema mejora la eficiencia a la hora de gestionar las alarmas del SCADA.

- Con el Sistema Experto el tiempo de respuesta del operador a la hora de dar solución a las alarmas es mucho que el que utiliza actualmente.

4. SOPORTE

- El sistema debe ser de fácil, configuración y puesta en marcha.
- La arquitectura debe ser abierta y modular, de capacidad escalable y tecnología actualizable de acuerdo a las necesidades operacionales y tendencias tecnológicas de las aplicaciones y componentes que se ejecutan o interactúan con el sistema de manera que permite implementar cambios, ya sea cualquier corrección, mejora o adaptación del sistema.

5. RESTRICCIONES DE DISEÑO

- El sistema debe ser multiplataforma: El sistema debe ejecutarse en diversas plataformas de hardware y software independientemente del número y composición de los bits de la arquitectura.
- Empleo del mismo conjunto de herramientas en el desarrollo del sistema: Se debe emplear el mismo conjunto de herramientas para el desarrollo del sistema, así como para futuros desarrollos. De manera tal que garantice la escalabilidad y compatibilidad entre los diferentes desarrollos.

6. INTERFAZ

INTERFAZ DE USUARIO

- La interfaz de usuario debe ser sencilla y amigable para permitir al operador una rápida y cómoda interacción con las funcionalidades del módulo.

INTERFAZ DE SOFTWARE

- Se debe tener instalado el SCADA.

INTERFAZ DE HARDWARE

- Familia de microprocesadores Intel Pentium Dual Core o superior.
- La memoria RAM igual o superior a 2048 MB.
- Disco Duro de 160GB de almacenamiento o superior.
- Arquitectura de 32 bit.

2.3 CONEXIÓN CON EL SCADA.

El en módulo BDTR del SCADA se almacena toda la información procesada en tiempo real, de este se obtendrán los datos necesarios de las alarmas que se activan en el sistema para su posterior procesamiento.

El Middleware como módulo de comunicación dentro del SCADA cuenta con una serie de tópicos o canales mediante los cuales recibe o brinda diferentes tipos de información, ya sea sobre los eventos, puntos o comandos del sistema. Para la obtención de la lista de alarmas a procesar es necesario subscribirse al Middleware, el cual realiza la petición y obtiene la información de las alarmas proveniente de la BDTR y la envía al módulo de IA para su tratamiento.

2.4 PROPUESTA DEL SISTEMA.

El sistema constará con una interfaz fácil de manejar que le permitirá al operador la configuración, visualización, y control de la base de conocimientos, la cual estará constituida por las reglas configuradas por un experto en el sistema. Esta interfaz le posibilitará al operador:

- Configurar la base de conocimientos del Sistema Experto de una manera fácil y entendible.
- Obtener un consejo de acción a ejecutar para dar respuesta a cada alarma activada en el sistema SCADA UX.
- Visualizar de una forma más comprensible las alarmas del SCADA UX.

Para que estas funcionalidades puedan desarrollarse la aplicación utilizará un Sistema Experto basado en Reglas, mediante las cuales se guardaran en la Base de Conocimiento, las posibles situaciones en las cuales se puede activar una alarma en el sistema (definidas previamente por un experto) y de acuerdo a ello, la solución que se le puede dar a la misma.

Como se mencionó en el capítulo anterior, un Sistema Experto tiene básicamente dos partes: la base de conocimiento y la máquina de inferencia. La Base de conocimientos estaría conformada por el sistema de reglas correspondientes con la configuración del sistema. La máquina de inferencia son los algoritmos que manejan la base de conocimiento para obtener conclusiones y propagar el conocimiento. Ahora bien, como el presente Sistema Experto es de tipo determinista, mucho menos complejo que el probabilístico, se tratará la inferencia mediante un algoritmo de inferencia, el cual logrará la conexión entre la base de conocimientos y la base de hechos del sistema, obteniendo las conclusiones aplicando la lógica de las reglas y realizando el proceso de razonamiento.

2.5 DESCRIPCIÓN DE LA BASE DE CONOCIMIENTO.

Las reglas de la Base de Conocimientos utilizarían el patrón SITUACIÓN-ACCIÓN, donde la situación estaría constituida por las diferentes situaciones en las que se puede activar una alarma en el sistema, comprobando de esta, el tipo de alarma activada, la causa de su activación, el recurso que la activó, el grupo al cual pertenece dicho recurso, la severidad y prioridad de la alarma; y la acción sería la respuesta a dar por parte del operador del sistema para el tratamiento de dicha alarma.

ARQUITECTURA DE LAS REGLAS.

Las reglas de la base de conocimientos podrán ser creadas a gusto del experto en el sistema, mientras que estas cumplan con las especificaciones del sistema y estén orientadas a solucionar las posibles alarmas que puedan ser generadas por el SCADA. Dichas reglas podrán tener desde 1 hasta 6 parámetros, cada uno de los cuales con su respectivo valor, además de estos, tendrá una respuesta en cada regla, la cual el experto definirá según las características de la regla. Dicha regla contará también con operadores (AND, OR) que posibilitará definir distintas posibles situaciones en una misma regla. A continuación un ejemplo de estas posibles reglas:

IF Tipo de Alarma = Alarma de Nivel AND

Causa = Bajo-Bajo OR

Tipo de Alarma = Alarma de Desviación OR

Prioridad = 2 AND

Severidad = Crítica AND

Recurso = Analógico2

THEN Respuesta: Informar al personal de mantenimiento sobre posible problema de funcionamiento en el recurso Analógico2.

2.5.1 El Proceso de Razonamiento.

Para el presente sistema de reglas trabajaremos con encadenamiento hacia delante o forward chaining ya que no sabemos con cuanta información contaremos para trabajar, por lo que hay que hacerlo con la menor posible.

PASOS A REALIZAR EN EL ENCADENAMIENTO HACIA DELANTE.

➤ **Matching.**

Búsqueda de las reglas para las que es cierto su antecedente.

➤ **Resolución de conflictos.**

- Selección, entre las reglas encontradas no redundantes, de aquella que se va a ejecutar
- Criterios de selección (*estrategia de búsqueda*):
 - Mayor nº de premisas en el antecedente.
 - Prioridad más alta.
 - Búsqueda en profundidad.

➤ **Ejecución.**

- Se dispara la regla, por lo que ampliamos los datos conocidos.

Un motor de inferencia podría ser aplicable, pero la incorporación de este al presente sistema experto sería innecesaria ya que según las características del sistema un motor de inferencia como Amzi Prolog, ExpertCoder, Jchell o CLIPS brindarían funciones que no son necesarias para la solución del presente problema, provocando esto, un tamaño excesivo en la aplicación y una mayor complejidad de procesamiento innecesaria para resolver un mismo problema, el cual puede ser resuelto con un algoritmo de inferencia. Se busca también que la aplicación sea independiente completamente de programas ajenos al SCADA.

2.5.2 Algoritmo de Inferencia.

Para el tratamiento de la inferencia en el Sistema para la Gestión de las Alarmas en el SCADA UX es utilizado el módulo QtScriptTools del Framework QT. Este permite utilizar la clase QScriptEngine, que evalúa el código script mediante la función evaluate(). Las expresiones a evaluar se crean realizando una comparación entre los valores de cada regla guardada en el sistema y cada alarma activada en el SCADA, utilizando los operadores AND y OR de dichas reglas para unir los predicados de cada expresión. Una vez que se obtiene la respuesta para cada expresión es posible darle solución a una alarma específica del sistema si existe una regla en la Base de Conocimiento que satisface las condiciones de activación de la alarma, en caso contrario no es posible darle solución a la alarma. Para escoger de entre todas las reglas que pueden darle solución a una alarma, se verifica entre estas la de mayor prioridad y esta es la seleccionada para dar respuesta a la alarma.

2.6 MODELO DE CASOS DE USO DEL SISTEMA.

2.6.1 Actores del sistema.

Actores	Justificación
SCADA	Este en el módulo o sistema externo que interactúa con el

	componente, y es el encargado de proveer las alarmas.
Operador	Es el que se encarga de configurar las reglas que posteriormente serán utilizadas para el tratamiento de las alarmas. Además de trabajar obtener el listado de mensajes visualizados por el sistema.

Tabla 1. Actores del Sistema.

2.6.2 Casos de uso del sistema.

Los casos de uso del sistema serán:

1. Generar Respuesta de Alarma.
2. Gestionar Reglas.

2.6.3 Diagrama de casos de uso (CU) del sistema.

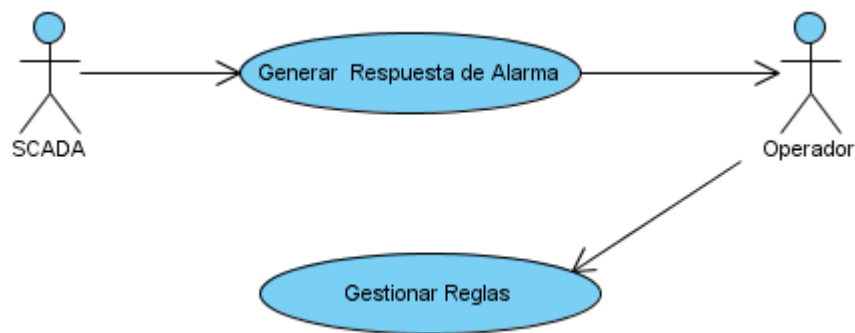


Fig. 4 Diagrama de CU.

2.6.4 Descripción de los casos de uso en formato extendido.

Caso de Uso	
CU-1	Generar Respuesta de Alarma.
Propósito	Permitir cargar y darle solución a las alarmas.
Actores: SCADA	
Resumen: El CU comienza cuando el SCADA obtiene la primera lista de alarmas, este a su vez provee estas alarmas para que nuestro sistema las procese y lograr la obtención de las posibles respuestas que se le pueden dar a cada una de ellas.	
Referencias	RF1, RF.1.1, RF2, RF2.1, RF2.2, RF3, RF3.1, RF3.2.
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	<ol style="list-style-type: none"> 1. Carga las alarmas generadas en el SCADA. 2. Procesa una a una según el orden establecido. 3. Extrae de la alarma en proceso el recurso que genera la alarma, la causa, el tipo, severidad, prioridad y el grupo operacional al cual pertenece el recurso que la generó. 4. Busca en la base de conocimientos, la regla que corresponda a esa alarma según los datos antes obtenidos de ella y la prioridad establecida en cada una de las reglas. 5. Extrae de la regla encontrada la solución para la alarma en proceso. 6. Manda un mensaje con la solución de la alarma a la cola de mensajes a visualizar. 7. Visualiza al operador la cola de mensajes de solución para las alarmas procesadas, 8. Termina el caso de uso.
Precondiciones	<ul style="list-style-type: none"> ➤ El SCADA ha sido instalado. ➤ La Base de conocimientos ha sido configurada por el operador.
Poscondiciones	<ul style="list-style-type: none"> ➤ Se muestra al operador la lista de mensajes con las soluciones a

	las alarmas.
Prioridad	Crítico.

Tabla 2.Descripción del CU Generar Respuestas de Alarma.

Caso de Uso	
CU-2	Gestionar reglas.
Propósito	Permitir configurar la Base de Conocimientos.
Actores: Operador	
Resumen: El CU comienza cuando el operador experto configura todas las reglas con la que contará la Base de conocimientos para tratar todas las posibles alarmas que podrá generar el sistema.	
Referencias	RF4, RF4.1, RF4.1, RF4.3
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Se accede al entorno para administrar las reglas.	1.1 Muestra las siguientes opciones. 1.2 Adicionar Regla, el sistema irá a la sección Adicionar Regla. 1.3 Modificar/Eliminar Regla, el sistema irá a la sección de Reglas.
Sección Adicionar Regla	
Acción del Actor	Respuesta del Sistema
2.1.1 Selecciona la opción de Adicionar Regla.	2.1.2 Permite adicionar una nueva regla.
2.1.3 Crea la nueva regla en la base de conocimientos.	2.1.4. Muestra al operador como quedará la regla a adicionar.
2.1.5 Acepta crear la nueva regla.	2.1.6 Verifica si la nueva regla no está registrada, o si tiene errores de sintaxis, sino es así, la guarda en la Base de conocimientos, terminando así el caso de uso.

Flujo Alternativo del Paso 2.1.6	
Acción del Actor	Respuesta del Sistema
	2.1.6a Verifica si la nueva regla no está registrada, si lo está, emite un mensaje de error al operador "Regla ya registrada".
	2.1.6b Verifica si la sintaxis de la nueva regla registrada es correcta, si no lo es, emite un mensaje de error al operador "La sintaxis de la regla es incorrecta"
	2.1.6c Verifica si la nueva regla cuenta con una respuesta, sino la tiene, emite un mensaje de error al operador "La regla debe tener una respuesta"
	2.1.7 Mantiene su configuración antes de la última operación.
Sección Reglas	
Acción del Actor	Respuesta del Sistema
2.2.1 Selecciona la opción Modificar/Eliminar Regla.	2.2.2. Muestra una lista de todas las reglas configuradas en la base de conocimientos. Y le muestra dos opciones, Modificar o Eliminar.
2.2.3. Selecciona una regla de la lista y escoge la opción Eliminar.	2.2.4. Borrará la regla de la base de conocimientos.
Flujo Alternativo del paso 2.2.3	
Acción del Actor	Respuesta del Sistema
2.2.3a. Selecciona la opción de modificar regla.	2.2.4a Le permite modificar las reglas.
2.2.5 Modifica los valores originales de la regla.	2.2.6 Verifica si la regla modificada no está registrada en la Base de conocimiento, sino está presente, guarda los cambios, terminando así el caso de uso.
Flujo Alternativo del Paso 2.2.6	
Acción del Actor	Respuesta del Sistema

	2.2.6a Verifica si la regla modificada está registrada en la Base de conocimiento, si lo está emite un mensaje de error al operador “Esta regla ya existe”.
	2.2.7 Mantiene la configuración original en la Base de conocimientos.
Precondiciones	<ul style="list-style-type: none"> • El SCADA ha sido instalado.
Poscondiciones	<ul style="list-style-type: none"> • Queda configurada la Base de Conocimientos, con las reglas a utilizar para el tratamiento de las alarmas.
Prioridad	Crítico.

Tabla 3. Descripción del CU Gestionar Reglas.

2.6.5 Diagramas de clases.

2.6.5.1 Diagrama de Clases de Análisis:

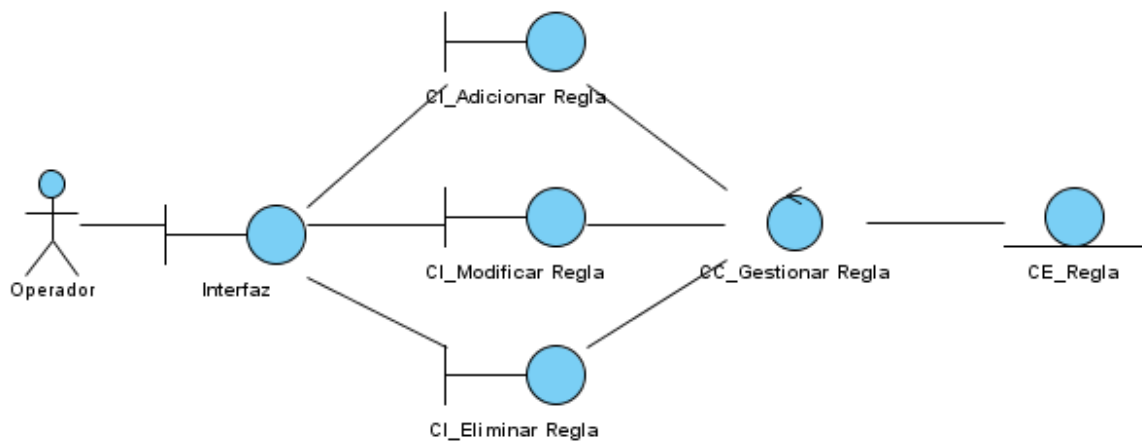


Fig. 5 Diagrama de Clases de Análisis. Gestionar Regla.

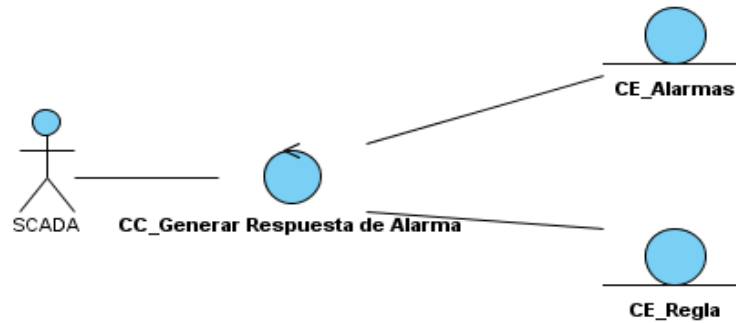


Fig. 6 Diagrama de Clases de Análisis. Generar Respuesta de Alarma.

2.6.5.2 Diagrama de Clases del diseño.

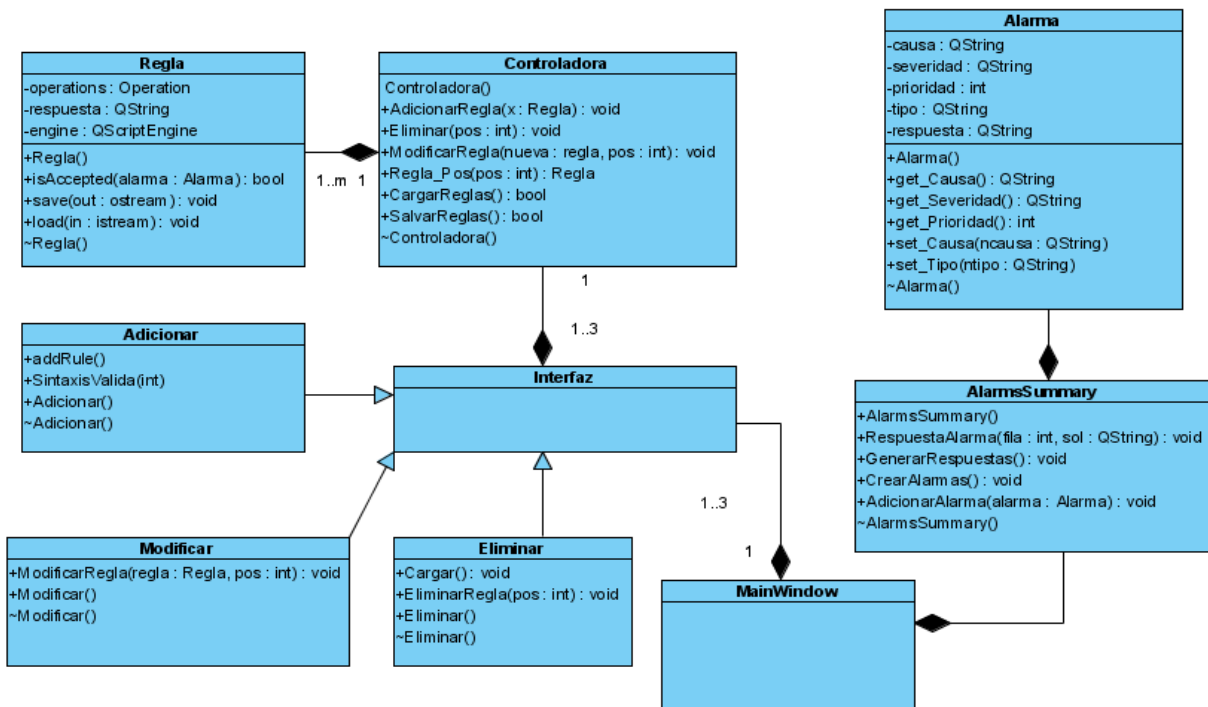


Fig. 7 Diagrama de Clase del Diseño General.

2.5.5.3 Diagramas de Secuencia.

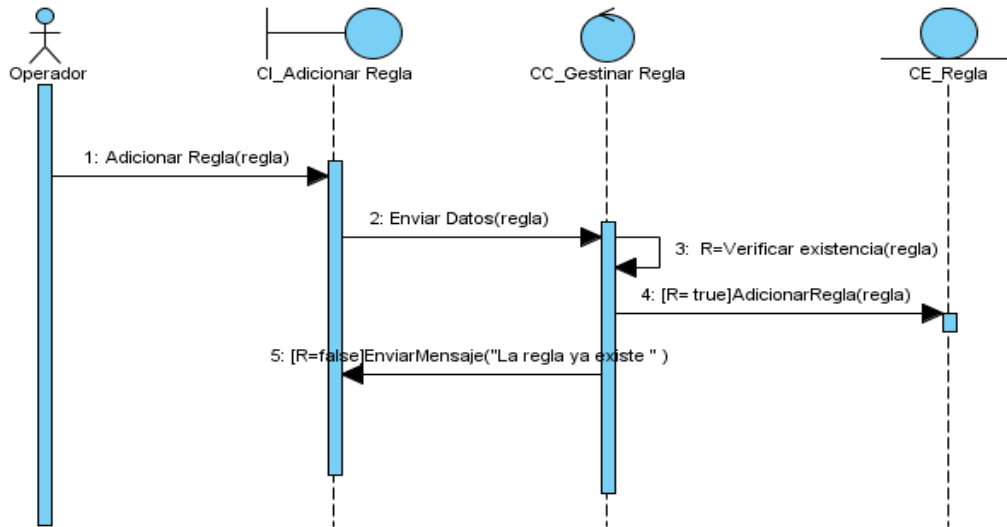


Fig. 8 Diagrama de Secuencia. Adicionar Regla.

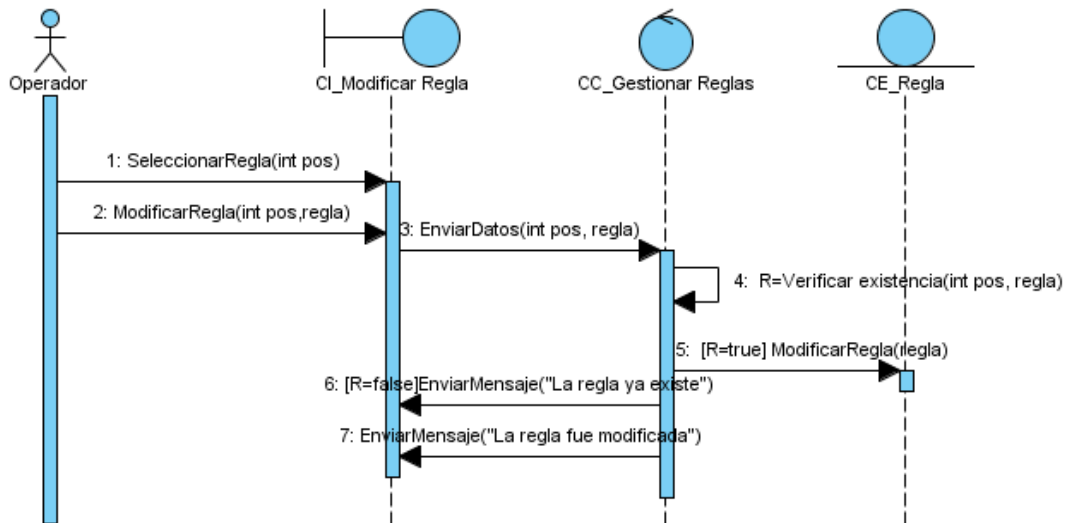


Fig. 9 Diagrama de Secuencia. Modificar Regla

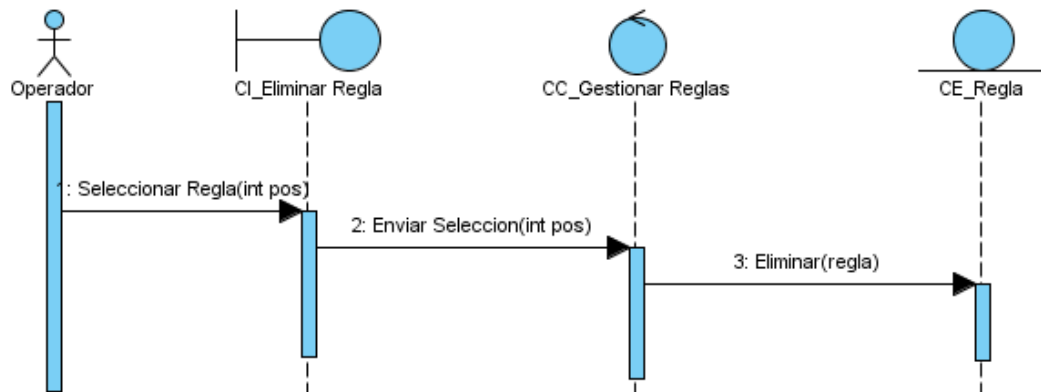


Fig. 10 Diagrama de Secuencia. Eliminar Regla

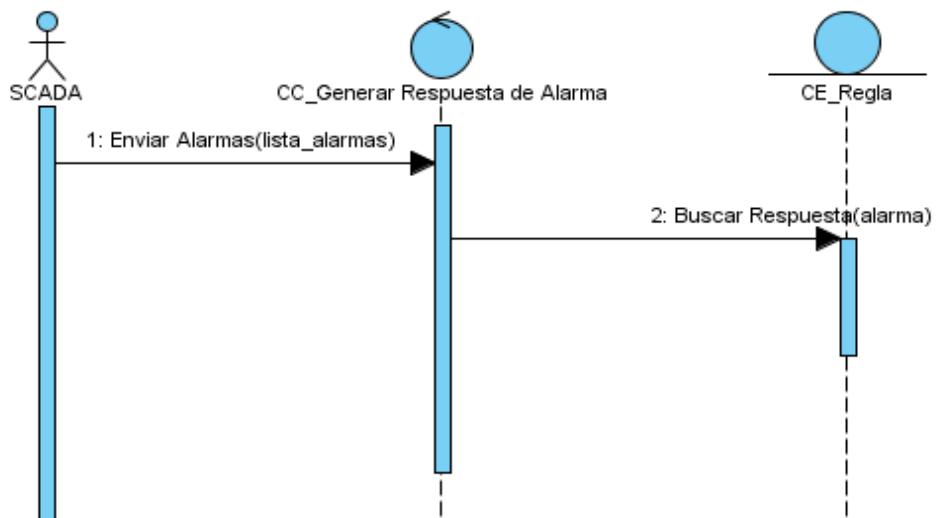


Fig. 11 Diagrama de Secuencia. Generar Respuesta de Alarma.

CAPÍTULO 3: VALIDACIÓN

En estos momentos el SCADA UX no tiene incluido ningún sistema de IA que le permita agilizar los procesos que este controla, por lo que el presente demo es el primer paso a la vinculación de la IA a estos sistemas, permitiendo de esta forma que el scada tenga una nueva funcionalidad y a su vez le da más valor agregado al producto, optimizar el tratamiento de las alarmas en el sistema.

En el presente capítulo se muestran imágenes del demo que servirá como referencia para el futuro desarrollo del módulo de IA y la vinculación de este con el SCADA de la facultad 5. Este demo al estar construido utilizando como guía los artefactos ingenieriles del capítulo 2. "Propuesta de Solución", sirve también de validación para demostrar la eficacia de dicha arquitectura. Para obtener una mayor seguridad con respecto a la eficiencia del diseño propuesto, se le realizan pruebas de Caja Negra a la aplicación demostrativa, con el objetivo de comprobar que se cumplan todos los requisitos funcionales propuestos.

3.1 DIAGRAMA DE COMPONENTES.

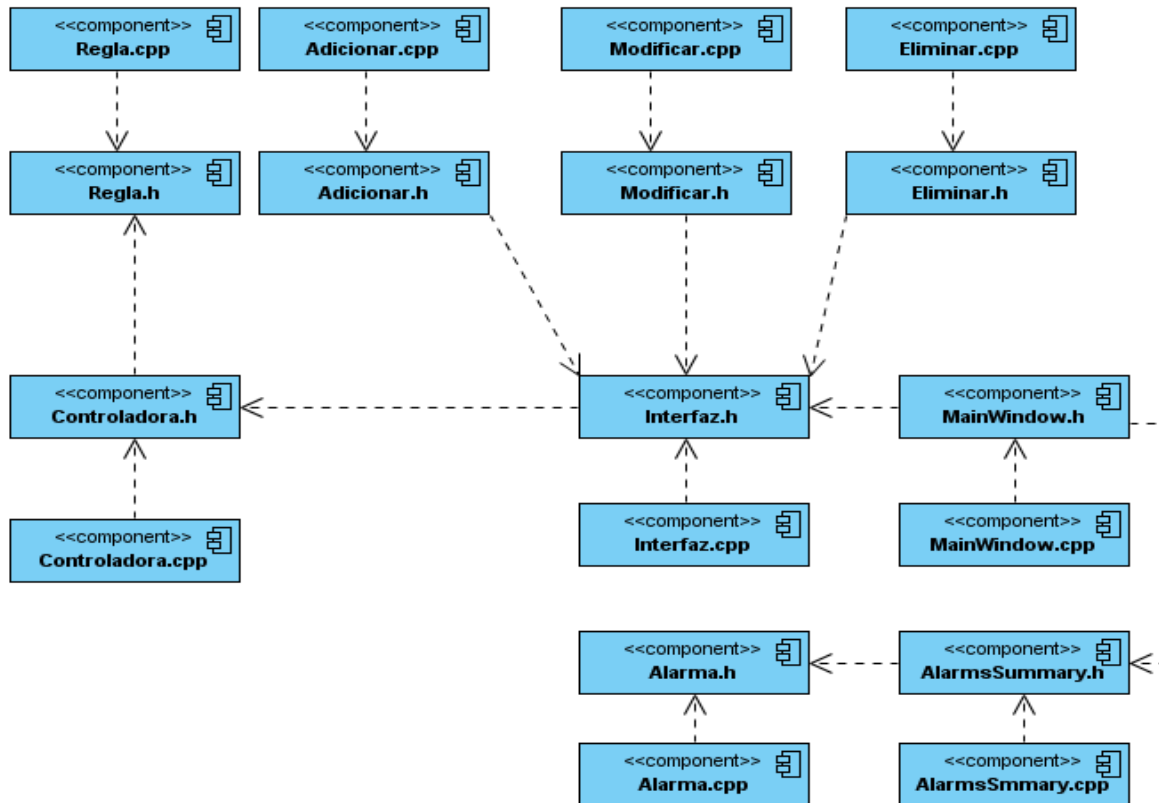


Fig. 12 Diagrama de Componentes.

3.2 FORMULARIOS DE LA APLICACIÓN.

En el formulario “Principal” el experto podrá escoger que acción realizará, esta pudiera ser “Adicionar Regla”, o “Modificar/Eliminar Regla”.

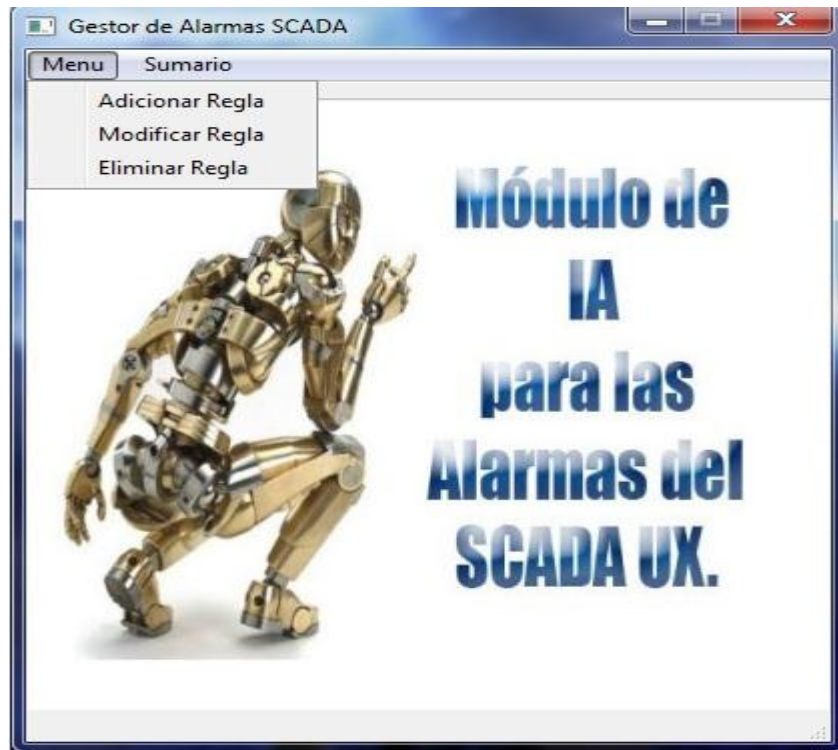


Fig. 13 Formulario Principal

Si el experto seleccionara la opción de “Adicionar Regla” del formulario “Principal” pasaría directamente al formulario de “Adicionar”, mediante el cual podrá agregar reglas a la base de conocimientos, esta acción preferiblemente se llevaría a cabo antes de poner en marcha el SCADA, ya que sin estas reglas, el módulo de IA no podría gestionar las alarmas generadas por el sistema, pero, puede darse el caso en que después de configurado el sistema de reglas existiese algún cambio en la industria o empresa y las especificaciones del SCADA cambiarían, de igual forma se podría adecuar la base de conocimientos para las nuevas especificaciones. Estas reglas se definen de acuerdo al sistema que será controlado por el SCADA, ya que al ser este, un sistema genérico, adaptable para cualquier tipo de industria o empresa que necesite la supervisión y control de los procesos que en ella se realicen, el sistema de reglas con que trabajará esta aplicación será igualmente predefinido antes ponerlo en funcionamiento.

Gestor de Alarmas SCADA

Menu Sumario

Elementos de la Regla

Prioridad: 5

Condición	Valores	Operador
Tipo_de_Alarma	Alarma_de_Nivel	AND
Prioridad	2	OR
Tipo_de_Alarma	Alarma_de_Desviacion	AND
Recurso	Analogico5	AND
Severidad	Media	AND
Grupo	Zona_1	

THEN Acción:

Respuesta a la Alarma

Respuesta 1

Regla Creada

```
IF Tipo_de_Alarma = Alarma_de_Nivel AND IF Prioridad = 2 OR IF Tipo_de_Alarma = Alarma_de_Desviacion AND IF Recurso = Analogico5 AND IF Severidad = Media AND IF Grupo = Zona_1 THEN Respuesta: Respuesta 1
```

Adicionar

Fig. 14 Formulario Adicionar.

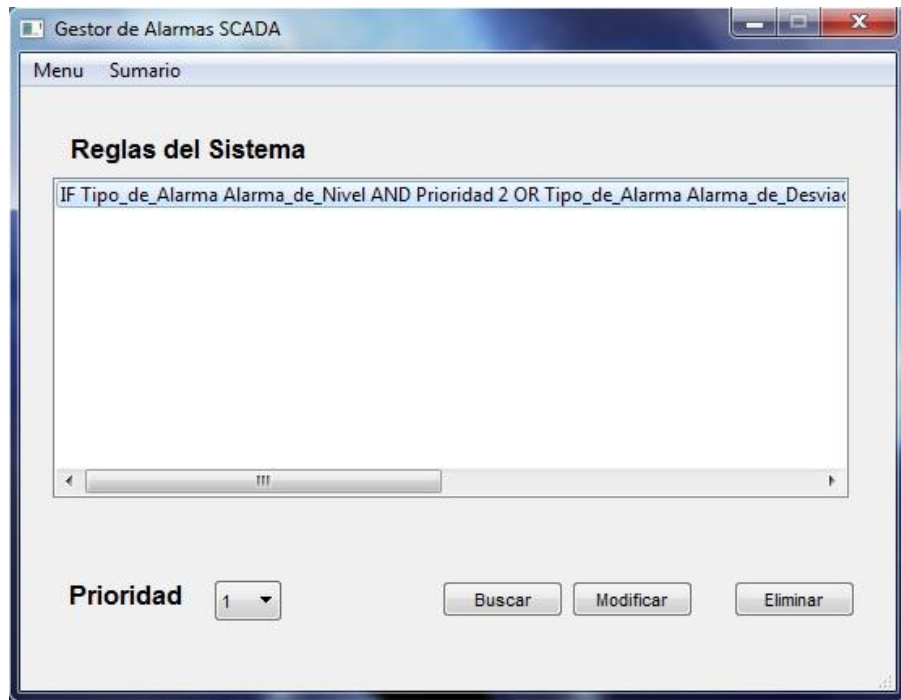


Fig. 15 Regla Adicionada.

Si el experto seleccionara la opción de "Modificar/Eliminar Regla", podría acceder al formulario "Reglas", el cual le permitiría seleccionar de la lista de reglas configuradas hasta el momento en la base de conocimientos, una regla a modificar o eliminar. Si desea modificar una regla, con seleccionarla en la lista y darle clic al botón "Modificar" de este formulario, iría directamente al formulario de "Modificar".

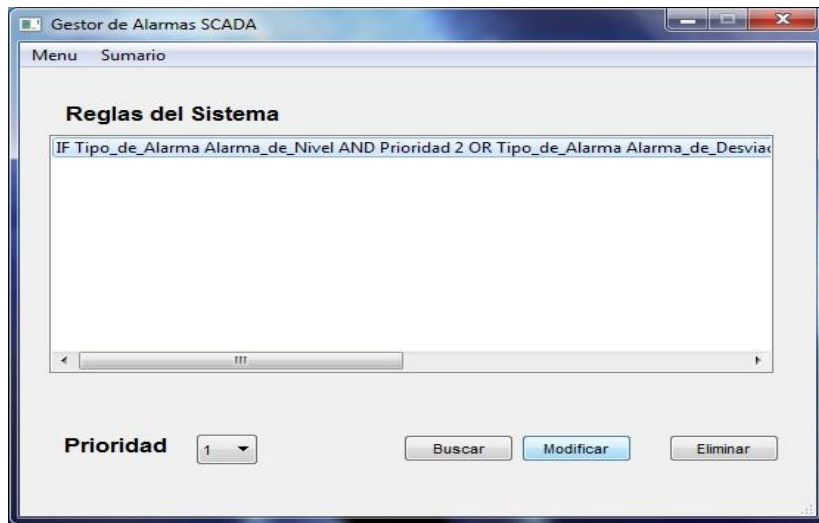


Fig. 16 Selección de la Regla a Modificar.



Fig. 17 Formulario Modificar con la regla cargada.

Por otro lado, si el experto desea eliminar completamente la regla de la base de conocimiento, igualmente con seleccionarla y accionar el botón “Eliminar”, esta sería borrada de la base de reglas.

Si el experto seleccionara la opción de “Modificar” en el formulario de “Modificar/Eliminar” se mostraría el formulario de “Modificar Regla”, el cual cargaría los valores de la regla seleccionada de la lista de reglas del formulario de “Reglas”, con este formulario el experto cambiaría los parámetros de la regla con sus respectivos valores, por otros más adecuados para lograr la gestión de las alarmas del sistema.

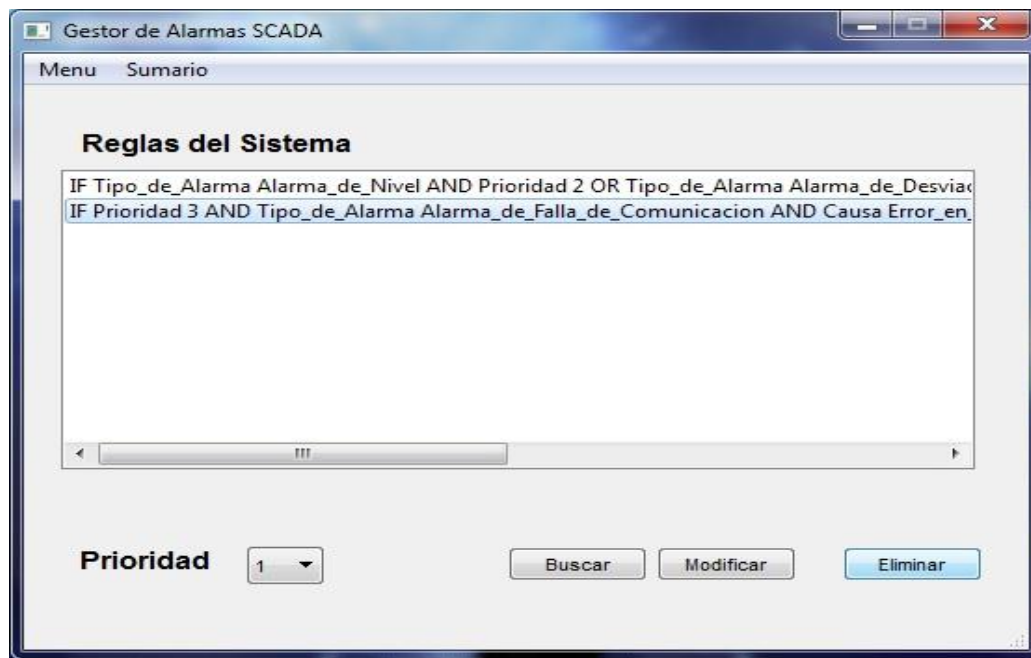


Fig. 18 Selección de la regla a eliminar

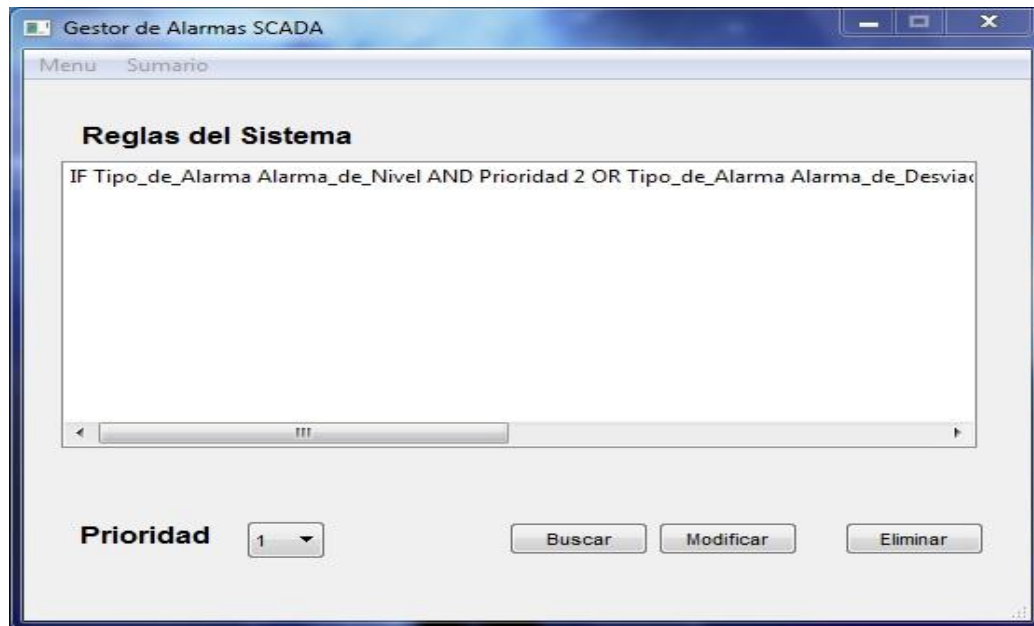


Fig. 19 Regla eliminada de la Base de Conocimientos.

Toda este Sistema Basado en Reglas se utilizaría para brindarle al operador del sistema una respuesta u opinión de un experto a la hora de tratar las alarmas del sistema SCADA. A continuación se presenta un ejemplo de cómo se visualizaría estas respuestas en un formulario con características similares el sumario de alarmas del sistema SCADA. Aquí se observan primeramente el sumario de alarmas sin haber utilizado el Sistema Experto y después de su utilización. Antes de utilizarlo el campo Respuesta del sumario aparece que con la palabra “respuesta” (Fig.20) y después de se puede apreciar que en cada alarma se visualiza la acción a realizar para darle solución a esta (Fig.21). Se puede ver también que se no se encuentra una regla que cumpla con las condiciones de la alarma, esta tendrá como respuesta “No solución”. En dicho caso hay que realizar cambios en la Base de Conocimientos y definir una o varias reglas para este tipo de alarmas.

The screenshot shows a 'Dialog' window with a table containing 5 rows of alarm data. The columns are: Tipo de Alarma, Causa, Severidad, Prioridad, Grupo, Recurso, and Respuesta. The 'Respuesta' column contains the word 'respuesta' for all entries.

	Tipo de Alarma	Causa	Severidad	Prioridad	Grupo	Recurso	Respuesta
1	Alarma de Tasa ...	Tasa de Cambio	Baja	1	Zona1	Analógico5	respuesta
2	Alarma de Nivel	Alto-Alto	Critica	5	Zona1	Analógico1	respuesta
3	Alarma de Cam...	Alarma de Cam...	Critica	5	Zona1	Analógico10	respuesta
4	Alarma de Nivel	Bajo	Media	3	Zona1	Analógico6	respuesta
5	Alarma de Nivel	Bajo	Media	3	Zona1	Analógico8	respuesta

Buttons at the bottom: OK, Cancel, Respuestas.

Fig. 20. Sumario sin Respuestas de Alarmas.

The screenshot shows a 'Dialog' window with a table containing 5 rows of alarm data. The columns are: Tipo de Alarma, Causa, Severidad, Prioridad, Grupo, Recurso, and Respuesta. The 'Respuesta' column contains specific responses for each entry.

	Tipo de Alarma	Causa	Severidad	Prioridad	Grupo	Recurso	Respuesta
1	Alarma de Tasa de...	Tasa de Cambio	Baja	1	Zona1	Analógico5	No solucion
2	Alarma de Nivel	Alto-Alto	Critica	5	Zona1	Analógico1	Avisar al equip...
3	Alarma de Cambi...	Alarma de Cam...	Critica	5	Zona1	Analógico10	Darle manteni...
4	Alarma de Nivel	Bajo	Media	3	Zona1	Analógico6	Comunicar fallas.
5	Alarma de Nivel	Bajo	Media	3	Zona1	Analógico8	Comunicar fallas.

Buttons at the bottom: OK, Cancel, Respuestas.

Fig. 21. Sumario con Respuesta de Alarmas.

3.3 DISEÑO DE CASOS DE PRUEBA.

3.3.1 Diseño de caso de Prueba.CU Gestionar Regla.

Descripción General:

El CU comienza cuando el operador experto se dispone a configurar las reglas que tendrá la Base de Conocimientos.

Condiciones de ejecución:

El sistema Scada debe haber sido instalado.

Secciones a probar en el caso de uso:

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 1. Adicionar Regla.	EC 1.1 Adicionar Regla exitosamente.	El operador selecciona la opción de Adicionar Regla. El sistema permite adicionar una nueva regla. El operador define los valores que tendrá la regla, y acepta crear la nueva regla, el sistema verifica si la nueva regla no está registrada y si no lo está la guarda en la Base de conocimientos	Se adiciona una nueva regla.
	EC 1.2 Adicionar una Regla que ya exista.	El sistema muestra un mensaje de error "Regla ya registrada".	Se cancela la operación de adicionar regla.
SC 2. Modificar Regla.	EC 2.1 Modificar Regla existente.	El operador selecciona la regla a modificar y la opción de "Modificar" El sistema le permite modificar las reglas luego el operador modifica los valores originales de la regla, el sistema verifica si la regla con los nuevos datos no es igual a ninguna de la base de conocimientos, sino es así, se guardan los cambios.	Se modifica una regla.
	EC 2.2 Al modificar la regla, ya se encuentra registrada.	El sistema muestra un mensaje de error "Regla ya registrada".	Se cancela la operación de Modificar Regla.

SC 3. Eliminar Regla.	EC 3.1 Eliminar Regla.	El operador selecciona la regla a eliminar y opción "Eliminar".	Se elimina la regla seleccionada de la base de conocimientos.
------------------------------	-------------------------------	---	---

Tabla 4 Diseño de Caso de Prueba.CU Gestionar Regla.

3.3.2 Diseño de caso de Prueba.CU Generar Respuestas de Alarma.

Descripción General:

El CU comienza cuando el SCADA obtiene la primera lista de alarmas, este a su vez provee estas alarmas para que nuestro sistema las procese y lograr la obtención de las posibles respuestas que se le pueden dar a cada una de ellas.

Condiciones de ejecución:

El Scada debe haber sido instalado y a su vez la base de conocimiento configurada por el operador.

Nombre de la Sección	Escenarios de la Sección	Descripción de la funcionalidad	Flujo Central
SC 2. Generar Respuesta de Alarma.	EC 2.1 Generar Respuestas de Alarmas.	<p>El SCADA envía la alarma o lista de alarmas generadas. El sistema:</p> <ul style="list-style-type: none"> • Carga las alarmas generadas en el SCADA. • Organiza según su prioridad y severidad. • Procesa una a una según el orden establecido. • Extrae de la alarma en proceso el recurso que genera la alarma, la causa, el tipo, severidad, prioridad y el grupo operacional al cual pertenece el recurso que la generó. • Busca en la base de conocimientos, la regla que corresponda a esa alarma según los datos antes 	Se le adiciona a cada alarma su respectiva respuesta.

		<p>obtenidos de ella y la prioridad establecida en cada una de las reglas.</p> <ul style="list-style-type: none"> • Extrae de la regla encontrada la solución para la alarma en proceso. • Manda un mensaje con la solución de la alarma a la cola de mensajes a visualizar. • Visualiza al operador la cola de mensajes de solución para las alarmas procesadas. 	
--	--	--	--

Tabla 5 Diseño de Caso de Prueba. CU Generar Respuesta de Alarma.

RESULTADOS DE LAS PRUEBAS.

Después de realizadas las pruebas de Caja Negra podemos concluir que el demo cumple con todos los parámetros definidos para el tratamiento de las Alarmas del SCADA. Estas pruebas arrojan resultados favorables donde se ve el cumplimiento de todos los requisitos funcionales establecidos logrando:

- La obtención de las alarmas.
- Asociar las reglas de la base de conocimientos con las alarmas del SCADA.
- Gestionar cada una de las alarmas del SCADA, adicionándole a estas una respuesta para lograr su tratamiento.
- Se permite la creación, modificación y eliminación de las reglas en la Base de Conocimientos.

CONCLUSIONES

Para el cumplimiento de los objetivos de este proyecto, teniendo en cuenta las características del sistema con el cual se trabajó y las herramientas y metodologías utilizadas para la creación del módulo de Inteligencia Artificial:

- Se propone como técnica a utilizar los Sistemas Basados en Reglas, ya que estas son las más propicias para resolver el problema existente en la gestión de las alarmas en el SCADA.
- Se obtienen los artefactos ingenieriles correspondientes a la etapa de análisis y diseño del módulo de IA para la gestión de las alarmas del SCADA UX.
- Se obtuvo una aplicación demostrativa en la cual se ve el funcionamiento del sistema basado en reglas.

RECOMENDACIONES

- Utilizar el presente trabajo como punto de partida para profundizar sobre estos temas en futuras investigaciones de Inteligencia Artificial específicamente en el trabajo de toma de decisiones.
- Valorar la integración de este módulo al Sistema SCADA UX para mejorar la supervisión y control de los procesos.
- Incluir en el módulo de IA la predicción de eventos a través de Sistemas Expertos, la cual permite acciones como la supervisión de señales, control inteligente, detección y diagnóstico de fallas.

BIBLIOGRAFÍA

1. **Febles Rodríguez, Dr. Juan P.** *La inteligencia artificial en la enseñanza de la medicina.* 2000.
2. **Freeman, J. A. y Skapura, DM.** *Redes Neuronales. Algoritmos, aplicaciones y técnicas de propagación.* Mexico : Addison-Wesley, 1993.
3. **Hilera González, José Ramón y Martínez Hernández, Victor Manuel.** *Redes neuronales artificiales: fundamentos, modelos y aplicaciones.* Madrid : RA-MA, 1995.
4. **Aspiazu, Guillermo Choque.** Ciencia y Computación. *Ciencia y Computación.* [En línea] 16 de Agosto de 2010. [Citado el: 5 de Febrero de 2011.] <http://www.eldiario.net/computacion/7-100816/index.htm>.
5. Vitutor. *Vitutor.* [En línea] 2010. [Citado el: 5 de Febrero de 2011.] http://www.vitutor.com/pro/2/a_17.html.
6. **Gregori, Fidel Aznar.** *Apuntes de Sistemas Expertos. Fundamentos de Inteligencia Artificial.* 2004.
7. **Bello Pérez, Rafael Esteban, y otros, y otros.** *Aplicaciones de la Inteligencia Artificial.* Jalisco : s.n., 2002. 970-27-0177-5.
8. **Huete, Juan Francisco.** *Sistemas expertos probabilísticos: modelos gráficos.* La Mancha : Cuenca : Servicio de Publicaciones de la Universidad de Castilla, 1998.
9. **Martínez, Irene y Rodríguez, Carmelo.** *Técnicas aplicadas al análisis de datos.* Almería : Almería: Servicio de Publicaciones de la Universidad de Almería, 2003.
10. **Penin, Aquilino Rodríguez.** *Sistemas SCADA 2da Edición.* Barcelona (España) : MARCONBO, 2007. ISBN: 978-84-267-1450-3.

11. **Booch, Grady, Jacobson, Ivar and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : ADDISON-WESLEY, 1999. ISBN 84-7829-036-2.
12. **Booch, Grady, Jacobson, Iver and Rumbaugh, James.** *El Lenguaje Unificado de Modelado.* s.l. : ADDISON-WESLEY, 2007. 978-84-7829-087-1.
13. *Introducción a la Arquitectura y Configuración del Guardian del ALBA.* Habana : s.n., 2008.
14. *Base de Datos de Tiempo Real en el Guardian del ALBA.* Habana : s.n., 2008.
15. *Introducción al Módulo HMI. Modo Ejecución.* **UCI.** Habana : s.n., 2008.
16. **Tapia, Farid Fleifel.** Monografias.com. [En línea] <http://www.monografias.com/trabajos16/la-inteligencia-artificial/la-inteligencia-artificial.shtml>.
17. **Krishnamoorthy y Rajeev, S.** *Artificial Intelligence and Expert Systems for Engineers .* s.l. : CRC; Bk&Disk edition, 1996. ISBN: 849391253.
18. **Cowell, Robert G, Dawid, Philip y Steffen L, Lauritz.** *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks (Information Science and Statistics).* s.l. : Springer. ISBN: 0387987673.
19. **H. Jaakkola, Y. Kiyoki, T. Tokuda.** *Information Modelling and Knowledge Bases XIX: Volume 166 Frontiers in Artificial Intelligence and Applications.* 2008. ISBN 1586038125.
20. **Iserlis, Yuri.** *Artificial Intelligence Around Us.* s.l. : Bookstand Publishing. ISBN 978-1-58909-668-4.
21. **Frederick Wilfrid Lancaster, Linda C. Smith.** *Artificial intelligence and expert systems.* 1992.
22. **Robert G. Cowell, Philip Dawid, Steffen L. Lauritzen, David J. Spiegelhalter.** *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks (Information Science and Statistics).* s.l. : Springer. ISBN: 0387987673.

23. **Vasarhelyi, Miklos , Bonson, Enrique y Hoitash, Rani.** *Artificial Intelligence in Accounting & Auditing.* Princeton : Markus Wiener Publishers, 2005. ISBN 1-55876-179-9.

24. *Artificial Intelligence.* New Delhi : s.n., 2008. ISBN 978-81-906566-6-5.

GLOSARIO

A

Algoritmo: Un conjunto de reglas bien definidas para la solución de un problema en un número finito de pasos.

Aplicación: En informática las aplicaciones son los programas con los cuales el usuario final interactúa, es decir, son aquellos programas que permiten la interacción entre el usuario y la computadora. Esta comunicación se lleva a cabo cuando el usuario elige entre las diferentes opciones o realiza actividades que le ofrece el programa.

Arco: Es la unión entre dos nodos y representa la dependencia entre dos variables.

G

Grafo: un grafo es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

Grafo acíclico: Grafo que no tiene ciclos.

H

Heurística: Regla que permite orientar un algoritmo hacia la solución de un problema. Técnica de programación que permite a un sistema la creación gradual de un valor óptimo para una variable específica por medio del registro de los valores obtenidos en operaciones anteriores. Técnica empleada en los sistemas de inteligencia artificial.

I

Inferencia: Una inferencia es una evaluación que se realiza entre conceptos que, al interactuar, muestran sus propiedades de forma discreta, que permitirá trazar una línea lógica de causa-efecto, entre los diferentes puntos inferidos en la resolución del problema.

Iteraciones: Número determinado de veces que se realiza un proceso.

M

Modelo probabilístico: Es un modelo que se basa en el cálculo matemático de probabilidades, o sea en el cálculo o determinación cuantitativa de la posibilidad de que se verifique un suceso.

N

Nodo: Un nodo es una variable aleatoria que puede tener varios estados, cada nodo será una estructura o registro que dispondrá de varios campos.

P

Programación evolutiva: La programación evolutiva (PE) es una rama de la computación evolutiva (o algoritmos evolutivos). La programación evolutiva es prácticamente una variación de los algoritmos genéticos, donde lo que cambia es la representación de los individuos. En el caso de la PE los individuos son tripletas cuyos valores representan estados de un autómata finito. Cada tripleta está formada por: *El valor del estado actual*un símbolo del alfabeto utilizado*El valor del nuevo estado*.

T

Técnica: Una técnica es un procedimiento o conjunto de estos, (reglas, normas o protocolos), que tienen como objetivo obtener un resultado determinado.