UNIVERSIDAD DE LA CIENCIAS INFORMÁTICAS

Facultad 4, Centro de Informática Industrial

Modelo de alertas sobre la ocurrencia de eventos extraordinarios en sistemas SCADA

Trabajo final presentado en opción al título de Máster en Informática Avanzada

Autor:

Ing. Luis Manuel Vidal Piña

Tutor:

Dr. Arturo César Arias Orizondo

La Habana, noviembre de 2017

Dedicatoria

A mi familia por su apoyo, a mis amigos que siempre están. A todos los profesores, tutores y especialistas que han contribuido a mi formación profesional. A Martha y su familia, que ya también es la mía.

Agradecimientos

Al Centro de Informática Industrial por el apoyo brindado para la realización de esta tesis, a José Antonio Aragón por su dedicación y efusividad a la hora de atender mis inquietudes, al profesor Arturo César y a todos los que de una forma u otra han brindado su apoyo incondicional para el desarrollo de esta investigación.

Declaración de autoría

Declaro por este medio que yo Luis Manuel Vidal Piña, con carnet de identidad 89020437400, soy el autor principal del trabajo final de maestría "Modelo de alertas sobre la ocurrencia de eventos extraordinarios en sistemas SCADA", desarrollada como parte de la Maestría en Informática Avanzada y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los ___ días del mes de _____ del año 2017.

_____

Ing. Luis Manuel Vidal Piña

Resumen

El constante aumento de la automatización de procesos de la industria ha traído aparejado el alto crecimiento del volumen de información que debe ser supervisada y controlada por los operadores de los sistemas de control, conocidos como sistemas SCADA. Partiendo de estudios que apoyan la teoría de que un operador no puede atender correctamente una alarma en menos de 10 minutos, se propone un modelo de alertas sobre la ocurrencia de eventos extraordinarios en sistemas SCADA, basado en técnicas de descubrimiento de conocimiento en bases de datos y minería de datos, que contribuye a aumentar la capacidad de los operarios para tomar decisiones y disminuir los contratiempos que un evento de esta magnitud puede implicar tanto en términos económicos como en pérdidas de vidas humanas. Se utilizan técnicas de minería de datos para el descubrimiento de patrones de forma automática, usando como fuente de datos las bases de almacenamiento de datos históricos (HDB) con que cuentan por lo general este tipo de sistemas, además se hace uso de la herramienta Weka y se analizan varios de sus algoritmos. El modelo propuesto estará orientado al Sistema de Automatización Industrial desarrollado en la Universidad de la Ciencias Informáticas, y puede ser integrado a otros sistemas con características similares. Los resultados obtenidos fueron evaluados y comparados con el funcionamiento del sistema arrojando resultados satisfactorios.

Palabras clave: minería de datos, modelo de alertas, sistemas de supervisión y control, SCADA

Abstract

The constant increase of the automation of processes of the industry has brought with it the high growth of the volume of information that must be supervised and controlled by the operators of the control systems, known as SCADA systems. Starting from studies that support the theory that an operator cannot properly handle an alarm in less than 10 minutes, a model of alerts is proposed on the occurrence of extraordinary events in SCADA systems, based on knowledge discovery techniques in databases and data mining, which helps to increase the capacity of operators to make decisions and reduce the setbacks that an event of this magnitude can imply both in economic terms and in loss of human lives. Data mining techniques are used for automatic pattern discovery, using as a data source the historical data storage bases (HDB) that usually have this type of systems, the Weka tool is also used and several of its algorithms are analyzed. The proposed model will be oriented to the Industrial Automation System developed at the University of Computer Science, and can be integrated to other systems with similar characteristics. The results obtained were evaluated and compared with the functioning of the system, yielding satisfactory results.

Keywords*: alerts model, data mining, SCADA, supervisory and control systems*

ÍNDICE

Introducción

El avance de las tecnologías de la información y las comunicaciones (TIC) ha contribuido al desarrollo y crecimiento de las diferentes ramas de la industria. La automatización de los procesos industriales ha permitido sustituir el trabajo manual, disminuir el margen de error provocado por el factor humano y por tanto agilizar su desarrollo.

Los avances tecnológicos de las últimas décadas han incidido fuertemente en el aumento de la productividad y aprovechamiento de los recursos, pudiéndose afirmar que casi el 100% de la producción industrial mundial es controlada por sistemas de supervisión y control automático o semiautomático (Wonderware, 2015).

En la Universidad de las Ciencias Informáticas (UCI), uno de los centros productivos es el Centro de Informática Industrial (CEDIN), el cual desarrolla soluciones encaminadas a facilitar la gestión, el control y la supervisión de los procesos asociados con la automatización industrial. En el CEDIN uno de los productos desarrollados es el Sistema de Automatización Industrial basado en GNU/LINUX (SAINUX), el cual cumple con varios de los estándares y competencias de los sistemas SCADA, y se continúa trabajando para elevar su competitividad y eficiencia con otros sistemas de procesamiento industrial.

Un sistema de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés), es una aplicación o conjunto de aplicaciones de software especialmente diseñada para el control de la producción, que se comunica con los dispositivos de campo y controla los procesos de forma automática desde la pantalla del ordenador. Este tipo de sistemas permite el acceso a diversos usuarios de toda la información que se genera en el proceso productivo, los que pueden tener distintos roles como: operadores, supervisores de control de calidad, mantenimiento, y otros (Cabús, y otros, 2004).

El concepto de SCADA se aplica prácticamente en todo proceso donde se exige un control de calidad, producción y optimización del servicio. Los sistemas de este tipo pueden aplicarse en diversas ramas de la industria, desde el control de tráfico, hasta el monitoreo, control y supervisión de plataformas petroleras.

El flujo principal de información en los sistemas SCADA lo constituyen las variables o puntos, ya que representan innumerables indicadores como son: presión, temperatura, flujo, potencia, peso, intensidad de corriente, voltaje, acidez(PH), densidad, carga, resistencia o capacitancia entre otros. Las variables son adquiridas mediante instrumentación o utilizando sensores conectados a autómatas o equipos de control.

Un concepto importante dentro de los sistemas de control industrial es el de alarmas, los cuales son eventos de tipo crítico, y pueden tener lugar producto a variaciones o tasas de cambio incorrectas, transiciones entre estados, valores fuera de rango de estos indicadores, así como

acciones indebidas del usuario, anomalías del sistema o de los dispositivos. Estas constituyen situaciones que requieren de especial atención del operador o de los sistemas automáticos por lo que deben presentarse como advertencias audibles y visibles además de ser almacenadas para su posterior uso.

La visualización de la información en los sistemas SCADA se realiza a través de los HMI (interacción hombre-máquina, por sus siglas en inglés). Son módulos de software que permiten la configuración de los recursos del sistema, la edición de interfaces gráficas de operación (despliegues o mímicos), así como el monitoreo y control del proceso.

Las alarmas son presentadas usualmente a través del HMI al operador por diferentes vías:

- Como indicaciones en los objetos gráficos que simulan el proceso.
- En sumarios organizados jerárquicamente en función de sus atributos (criticidad, estado, hora de ocurrencia).
- Como señales sonoras en función del estado de las alarmas.
- En históricos de ocurrencias de alarmas entre otros.

Según define la Asociación de Materiales de Usuario y Equipamiento Ingenieril (EEMUA 191, 2007), un operador está en capacidad de atender de forma coherente y efectiva un promedio de una alarma cada 10 minutos, o al equivalente de 144 alarmas diarias en el mejor de los casos. Sin embargo, existen centros de control de alarmas donde se registran alrededor de dos mil alarmas diarias. En casos más críticos, como en centros de despacho de energía eléctrica, se pueden tener hasta más de cuarenta mil alarmas en un día. Esta cifra es muy superior al promedio que un operador está en capacidad de atender con eficiencia y eficacia (O. Aizpurúa, R. Galán, A. Jiménez, 2010). El término utilizado para definir el arribo en tiempo real de un número elevado de alarmas relacionadas generalmente por una causa común o raíz, es usualmente conocido como avalancha de alarmas.

Una operación incorrecta de las alarmas puede ocasionar según (Zabre, y otros, 2012):

- Detrimento de la calidad de la producción.
- Disminución de la eficiencia del proceso.
- Pérdida irreparable del equipamiento técnico.
- Afectaciones medioambientales.
- Peligro para la integridad física del personal.

A pesar de que la aplicación de metodologías de configuración y racionalización de alarmas han demostrado una reducción significativa de la probabilidad y la magnitud de avalanchas de alarmas (Zabre, y otros, 2012), el problema no ha sido considerado resuelto en su totalidad. En un estudio

realizado mediante un análisis comparativo de metodologías, modelos y algoritmos existentes que abordan esta temática se evidencia la ausencia de una posición acertada a la hora de abordar la solución al problema (ANSI/ISA-18.2, 2009) (O. Aizpurúa, R. Galán, A. Jiménez, 2010) (Zabre, y otros, 2012):

- La mayoría de las soluciones expuestas en trabajos científicos son orientadas a sistemas eléctricos de potencia, dificultando su generalización en el control de otros procesos industriales.
- Es cuestionado en el ámbito académico la aplicabilidad de alguno de los métodos disponibles.
- En la mayoría de las investigaciones se asume que las avalanchas de alarmas son ocasionadas por solo una causa raíz.
- Existen productos comerciales que realizan gestión inteligente de alarmas, pero no muestran los mecanismos utilizados para lograr este objetivo.

El uso de sistemas expertos, redes neuronales, redes bayesianas, teoría de decisiones y algoritmos híbridos como los neuro-borrosos, son algunas de las soluciones propuestas, sin embargo, la problemática persiste, debido a que no existe una metodología disponible que integre las entidades y tome en cuenta todos los dominios que participan en el proceso de administración de alarmas.

A partir de la situación descrita se formula el siguiente **problema de investigación**: ¿Cómo contribuir a prevenir la ocurrencia de eventos de alta criticidad y concurrencia por los operadores de sistemas de supervisión y control de procesos industriales?

El **objeto de estudio** se centra en el análisis de eventos extraordinarios en los sistemas SCADA.

El **objetivo general** de la investigación consistió en: desarrollar un modelo de alertas basado en la extracción de conocimiento de bases de datos de históricos en sistemas SCADA, que ofrezca la posibilidad de prevenir eventos de carácter extraordinario a los operadores de los centros de control.

Este objetivo general fue desglosado en los siguientes **objetivos específicos**:

1. Construir el marco teórico y referencial de la investigación, relacionado con la gestión inteligente de avalanchas de alarmas en sistemas de supervisión y control de procesos.
2. Diagnosticar la situación que presentan los modelos o mecanismos para la prevención inteligente de alarmas en sistemas de supervisión y control de procesos.
3. Desarrollar un proceso de extracción de conocimiento de las bases de datos del SAINUX a partir de su integración con herramientas de minería de datos.
4. Diseñar un procedimiento para la aplicación del modelo definido.
5. Integrar el modelo desarrollado al SAINUX.
6. Validar el modelo propuesto a través de los métodos definidos en la investigación.

Campo de acción: Modelos de alertas para el análisis de eventos de alta criticidad y concurrencia en sistemas SCADA.

Para dar solución al problema planteado y como resultado de la consulta y análisis de la literatura especializada, se formula la siguiente **hipótesis**: Con el desarrollo de un modelo de alertas basado en la extracción de conocimiento de bases de datos de históricos en sistemas SCADA, se contribuirá a prevenir por los operadores la ocurrencia de eventos de alta criticidad y concurrencia.

Para cumplir con las expectativas de la investigación fueron aplicados los siguientes métodos y técnicas para el trabajo científico:

Métodos teóricos:

- **Hipotético - deductivo**: Para formular la hipótesis de la investigación y proponer nuevas líneas de trabajo a partir de los resultados.
- **Analítico - sintético**: Para descomponer el problema de investigación en elementos, profundizar en su estudio y sintetizarlos en la solución propuesta.
- **Histórico - lógico**: Con el fin de realizar un estudio crítico sobre la evolución de los diferentes enfoques relativos a técnicas y modelos de racionalización de alertas y el control inteligente de alarmas.
- **Modelación**: para la representación explícita de la solución propuesta mediante la creación del modelo de alertas sobre la ocurrencia de eventos extraordinarios en sistemas de supervisión y control de procesos industriales, así como las ideas y los referentes teóricos extraídos de las fuentes bibliográficas consultadas.
- **Sistémico**: Para integrar armónicamente los componentes del modelo desarrollado, el proceso de descubrimiento de conocimiento y el sistema de automatización con que se cuenta (SAINUX).

Métodos empíricos:

- **Análisis documental**: En la revisión de la literatura especializada, tanto académica como empresarial, para extraer la información necesaria que permite realizar el proceso de investigación.
- **Análisis comparativo**: Para detectar similitudes, diferencias e insuficiencias en modelos, estrategias e investigaciones similares.
- **Observación participante**: Para el seguimiento de la actividad científica de los principales investigadores en el área del control inteligente de alarmas.

- **Likert, Iadov**: Para evaluar y corroborar por expertos y potenciales usuarios, la factibilidad y pertinencia del modelo.

- **Métodos estadísticos**: Realización de estudios experimentales y mediciones relacionadas con los componentes del modelo propuesto.

La novedad de la investigación se expresa mediante los siguientes aportes.

Aportes prácticos:

- Una aplicación para la extracción, transformación y procesamiento de los registros almacenados en el SCADA-SAINUX.
- Una herramienta para la clasificación de instancias de eventos extraordinarios a partir de la base de casos que constituye la Base de Datos Históricos del SCADA-SAINUX.
- Una herramienta para el agrupamiento de las instancias de eventos extraordinarios y su representación gráfica.
- Una herramienta para la extracción de reglas de asociación de eventos extraordinarios y su exportación en un documento textual.

Aporte económico-social:

- Constituye una alternativa a modelos propietarios de difícil acceso para las organizaciones cubanas que tienen dentro de sus líneas de trabajo el desarrollo de sistemas de supervisión y control de procesos.

- Su aplicación permite elevar la competitividad en el mercado internacional de los productos que desarrolla el Centro de Informática Industrial.

La tesis estará estructurada en: introducción, tres capítulos, conclusiones, recomendaciones, bibliografía y un cuerpo de anexos.

Capítulo I: Dedicado al análisis de los elementos esenciales que contribuyen al **fundamento teórico** del modelo de alertas sobre la ocurrencia de eventos extraordinarios en sistemas SCADA.

Capítulo II: Aborda la concepción y desarrollo de la **propuesta de solución**, en este caso del modelo. Para ello se caracteriza la situación inicial a partir del análisis comparativo entre modelos y metodologías existentes. Se selecciona la metodología para el desarrollo del modelo y se diseña el modelo definiendo sus principios, componentes, estructura y funcionamiento. Por último, es descrita la herramienta informática desarrollada para soportar el modelo.

Capítulo III: Contiene la **evaluación de los resultados** alcanzados en la investigación, los que permitieron confirmar la factibilidad, pertinencia y contribución del modelo propuesto a la resolución del problema planteado.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se analizan investigaciones, teorías, enfoques y antecedentes de gran validez para el desarrollo de la investigación. Se realiza un estudio para caracterizar los modelos de alertas usados en los ambientes de supervisión y control de procesos a escala industrial, de igual forma se incluye un análisis del o SAINUX y de los Sistemas SCADA en sentido general. También se incluye una investigación sobre las técnicas y algoritmos empleados para establecer tendencias a partir de los datos. Se realiza además un análisis comparativo entre las metodologías y herramientas existentes para el trabajo con algoritmos y modelos para el descubrimiento de conocimiento en bases de datos. Finalmente se analizan y elijen los estándares de intercambio de información entre aplicaciones para su uso en la comunicación entre la aplicación desarrollada y el sistema SCADA

1.1 Sistema de Automatización Industrial

El Sistema de Automatización Industrial basado en tecnologías GNU/LINUX conocido como SAINUX desarrollado en el Centro de Informática Industrial de la Universidad de las Ciencias Informáticas, es un software contemplado dentro de la gama de los sistemas SCADA. Los SCADA son sistemas basados en computadoras que permiten supervisar y controlar a distancia una instalación de cualquier tipo, donde el lazo de control es generalmente cerrado por el operador (entiéndase que requiere la intervención y/o supervisión de este). Proporciona comunicación con los dispositivos de campo (controladores autónomos, autómatas programables, etc.) y controla el proceso de forma automática desde la pantalla del ordenador. De igual forma, provee toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros supervisores dentro de la empresa: control de calidad, supervisión y mantenimiento (Karnouskos, 2011).

SCADA es la tecnología que permite a los usuarios recolectar la información de una o más instalaciones distantes, además de enviar instrucciones de control a esas instalaciones. Un SCADA hace innecesaria la presencia física de los operadores en los sitios distantes. Incluye interfaces para operadores y permite el monitoreo de los datos relacionados con los procesos (Boyer, 2004).

Se da el nombre SCADA 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 (Rodríguez Penin, 2012).

Los SCADA brindan una serie de posibilidades partiendo desde el control y supervisión de procesos en el nivel de campo hasta el soporte a los niveles de gestión y administración. Entre estos se encuentran los siguientes (Bailey, 2003), (Rodríguez Penin, 2007):

- Monitorización: realizan la representación de datos del proceso en tiempo real, es decir las variables que se leen de los dispositivos: temperatura, presión, etc.

- Supervisión remota: permite conocer el estado y desempeño del proceso desde estaciones centrales. Resulta muy útil en procesos distribuidos en amplias locaciones, y permite coordinar labores de control de calidad y mantenimiento.
- Control remoto de instalaciones y equipos: posibilita cambiar datos claves del proceso directamente desde el ordenador (abrir o cerrar válvulas, encender motores, etc). También permite ajustar los valores, parámetros y algoritmos de control.
- Visualización dinámica: genera imágenes dinámicas que representan de manera intuitiva el comportamiento del proceso, reflejando los elementos de la planta. En estos gráficos también se pueden encontrar curvas y tablas de los datos y estados del sistema en el tiempo.
- Adquisición y registro histórico de datos: los datos adquiridos son procesados en tiempo real y almacenados en bases de datos, de manera que puedan ser analizados posteriormente a fin de evaluar el desempeño del sistema, así como realizar el diagnóstico y prevención de fallas.
- Representación de señales de alarma: por medio de las señales de alarma el sistema informa al operador la presencia de una falla o condición indeseable en el proceso, estas señales pueden ser visuales y/o sonoras.
- Programación de aplicaciones: existe la capacidad de programar informes, estadísticas o recetas para los autómatas.
- Comunicación entre aplicaciones: permiten el intercambio de información con diversas aplicaciones.

Este tipo de sistemas pueden funcionar de manera centralizada o distribuida (característica que también comparte el SAINUX), para ello y una mejor distribución se agrupan las funcionalidades en módulos, los cuales pueden ser instalados en distintos nodos físicos por separado manteniendo una estrecha comunicación. La modularidad en estos sistemas posibilita trabajar con estándares arquitectónicos tales como bajo acoplamiento y alta cohesión entre otros. Entre los módulos se pueden mencionar (Montero, 2004), (Aragón Cáceres, 2011):

- Configuración: permite al usuario definir el entorno de trabajo de su SCADA, adaptándolo a la aplicación particular que se desea desarrollar.
- Interfaz Gráfica del Operador: proporciona al operador las funciones de control y supervisión de la planta. El proceso se representa mediante despliegues y gráficos generados desde el editor incorporado en el SCADA o importados desde otra aplicación durante la configuración del paquete.
- Módulo de procesamiento: ejecuta las acciones de mando pre-programadas a partir de los valores actuales de variables leídas.
- Gestión y archivo de datos: se encarga del almacenamiento y procesado ordenado de los datos, de forma que otra aplicación o dispositivo pueda tener acceso a ellos.
- Comunicaciones: se encarga de la transferencia de información entre la planta y la

arquitectura hardware que soporta el SCADA, y entre ésta y el resto de elementos informáticos de gestión.

En el caso particular del SAINUX, este software implementa más de 15 protocolos para la comunicación con los dispositivos especializados, entre los que se encuentran protocolos de los más extendidos como EthernetIP, ModbusRTU, ModbusELAM, ABEthernet y Dnp3 en disímiles variantes, entre otros. La arquitectura de este sistema es modular lo que permite que el despliegue del SAINUX pueda ser distribuido en servidores dedicados para el correcto balanceo de carga y rendimiento como se muestra en la Figura 1.

Las funciones que realizan cada uno estos módulos son las siguientes (Aragón Cáceres, 2011):

- Capa de comunicaciones o middleware: Es la capa que permite la comunicación entre todos los módulos del sistema.
- Adquisición y Procesamiento: Garantiza la ejecución en tiempo real y se encarga de la planificación de los procesos de lectura y escritura sobre los dispositivos. Incluye los drivers o manejadores que son los que permiten al sistema operativo interactuar, controlar y comunicarse con un dispositivo en particular, posibilitando la transmisión de datos entre redes de computadoras.
- Configuración: Es el encargado de almacenar, persistir y suministrar la información base para el funcionamiento de los demás módulos del SCADA.
- Visualización: Permite desde cualquier PC cliente acceder a los datos que se encuentran en el servidor, es la interfaz visual que muestra a los operadores el funcionamiento de los procesos de la planta. Contiene un subsistema de Reportes que permite emitir informes que consoliden la información adquirida.
- BD Históricos: Es el módulo encargado de manejar de forma clara, sencilla y ordenada los datos recolectados del campo que posteriormente se convertirán en información relevante y serán utilizados para realizar análisis y obtener retroalimentación de los procesos.
- Seguridad: Provee las funcionalidades necesarias para garantizar el trabajo autorizado por usuarios, además brinda las herramientas para la protección contra ataques maliciosos o involuntarios al sistema.
- Comunicación con terceros: Permite la comunicación con sistemas externos con el objetivo de intercambiar información. Estos terceros pueden ser otros sistemas SCADA, DCS, aplicaciones de gestión, gerenciales y de negocio.
- Planificador de tareas: Permite la programación y ejecución de scripts para realizar tareas que complementan el funcionamiento del sistema.

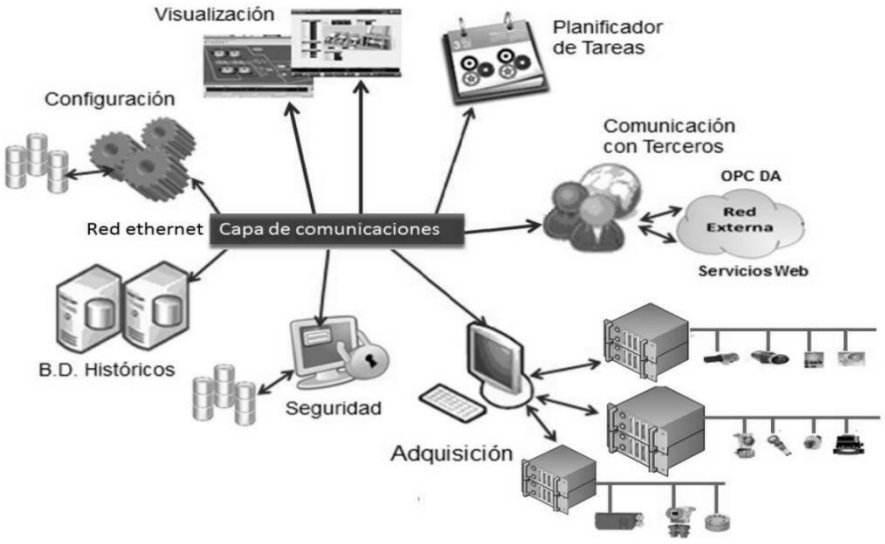


Figura 1. Arquitectura del Sistema de Automatización Industrial-SAINUX (Antúnez, 2015)

Entre los módulos más importantes se encuentra el de Base de Datos Históricos, encargado de almacenar la información concerniente a prácticamente todos los datos asociados con los procesos que controla el SAINUX y su persistencia en el tiempo. En estas bases de datos se almacenan datos como: características de los puntos y variables, variaciones importantes en el valor de puntos, registro de accesos al sistema, fallas, registro y descripción de anomalías, alarmas y otros muchos. Los datos asociados a este módulo perduran el tiempo previamente definido por los administradores del sistema, o en su defecto el sistema lo hará indefinidamente. Esta base de datos constituye una fuente de información que puede ser analizada y orientada al diseño y creación de modelos de inteligencia artificial, aprendizaje automático y minería de datos para la extracción de conocimiento oculto, y el posterior apoyo en el proceso de toma de decisiones.

Es importante señalar que el módulo de Base de Datos Históricos constituye uno de los elementos más comunes en los sistemas SCADA, por lo que el modelo propuesto es aplicable a todos los sistemas salvando las características técnicas necesarias, filosofía de trabajo etc.

1.2 Modelos de alertas en ambientes de supervisión y control de procesos

Como ya se ha mencionado anteriormente, existen una cantidad importante de sistemas que aplican técnicas de configuración y racionalización de alarmas, los cuales han mostrado resultados bastante significativos en cuanto a la reducción de la probabilidad de ocurrencias y la magnitud de las avalanchas de alarmas, no obstante el problema no se considera resuelto en su totalidad (Zabre, y otros, 2012). Partiendo sobre todo de que existen productos comerciales que realizan una gestión inteligente de las alarmas ocurridas en los sistemas que manejan, pero no publican los mecanismos utilizados para lograr este objetivo por tratarse de soluciones de sistemas privativos (O. Aizpurúa, R. Galán, A. Jiménez, 2010) (Zabre, et al., 2012). No obstante, a continuación, se procede a enunciar algunas de las técnicas usadas para resolver este tipo de problemas.

1.2.1 Modelos basados en similitud

Las técnicas basadas en similitud están dirigidas a reducir el número total de alertas haciendo uso del agrupamiento y agregación de las similitudes identificadas. Cada evento extraordinario generado tiene asociados varios atributos o campos como pueden ser: dirección IP de destino y dirección IP de salida, número de puertos de destino y salida, protocolos, descripciones, e información de estampa de tiempo (*timestamp*) entre otros. La premisa en este tipo de modelos es que los eventos extraordinarios con características similares tenderán a mostrar una causa raíz similar o efectos similares en los sistemas de supervisión y control de procesos (Salah, y otros, 2013). El cómo definir las medidas de similitud constituye un problema de rendimiento crítico aún en la actualidad, en este sentido varias medidas de similitud han sido propuestas por los investigadores con el objetivo de definir una medida de similitud coherente con cada atributo, y cada uno de estos atributos debe contar con una diferente ponderación y efecto en el proceso de correlación.

Los modelos y técnicas basados en similitud se dividen en dos grandes grupos: aquellos que se basan en la similitud de atributos y aquellos que se basan en la información temporal.

La **correlación basada en atributos** relaciona a los eventos extraordinarios teniendo en cuenta la similitud entre algunos de sus atributos o rasgos, como dirección IP de destino y dirección fuente, puertos, estampas de tiempo, tipo de servicios o usuarios entre muchos otros. Comúnmente una medida de similitud es calculada empleando determinadas métricas como las funciones de Distancia Euclidiana, Mahalanobis, Minkowski, y distancia de Manhatan. Los valores resultantes son comparados con valores de umbrales para determinar si los eventos en cuestión tienen correlación o no (Salah, y otros, 2013). Algunas de las principales contribuciones científicas muestran interesantes aportes como (Alfonso, y otros, 2001), donde se tiene pone en práctica el uso de métodos probabilísticos de correlación de alertas basados en un marco de trabajo (*framework*) matemático, el cual permite encontrar la especificación de similitud mínima fusionando los eventos extraordinarios de múltiples fuentes de datos. Un procedimiento similar se realiza en (Lee, y otros, 2008), teniendo esta investigación la característica de usar la distancia Euclidiana para calcular el valor de similitud entre dos eventos para luego realizar un agrupamiento. Esta técnica es usada para la detección de Ataques de Denegación de Servicios (DDOS).

Los **modelos basados en series temporales** son modelos de correlación que prestan especial atención al factor tiempo con el objetivo de obtener la relación entre los eventos y correlacionarlos. La idea principal consiste en cuales eventos causan las mismas fallas y observando estrechamente los espacios de tiempo entre los eventos. El método más simple en series temporales lo constituye el conocido como ventana de tiempo, donde solo se procede a correlacionar a los eventos que ocurren dentro de una misma ventana de tiempo. Su principal ventaja reside en reducir el número de eventos generados por los nodos de administración y la

conversión e interpretación como nodos de máxima alerta. Las relaciones más fuertes en este tipo de técnicas entre dos eventos son etiquetados como fuertes, o débil cuando el intervalo de tiempo no puede ser precisado, sin embargo estos enfoques son usualmente deterministas, lo cual limita su aplicabilidad (Salah, y otros, 2013). En publicaciones como en (Ahmadinejad, y otros, 2009), son usadas varias ventanas de tiempo para evitar la necesidad de comparar los nuevos eventos con todo el grupo de eventos archivados. Luego es aplicada una función de estimación para calcular el umbral para realizar la correlación. Dos eventos se consideran relacionados si su similitud temporal es mayor que el umbral de similitud obtenido. En otras como en (Qin, y otros, 2003), en lugar de usar ventanas de tiempo, es aplicado en método de análisis estadístico basado en series temporales para determinar que dos eventos tienen correlación o no.

1.2.2 Modelos basados en secuencias

En este tipo de modelo los eventos son relacionados analizando la causalidad entre su ocurrencia. Las precondiciones son definidas como los requisitos necesarios para que un evento tenga lugar y las consecuencias son definidas como los efectos que tienen lugar luego que un evento específico ha ocurrido. Esta relación generalmente es representada como una ecuación lógica usando combinación de predicados tales como los conjuntivos o disyuntivos (AND/OR). Una de las principales ventajas es su escalabilidad, poniendo al descubierto de forma potencial la relación entre eventos, siendo por tanto fácil de entender ante escenarios de inestabilidad en los distintos sistemas de supervisión y control. Sin embargo, también se han identificado algunas desventajas como la probabilidad de contener un gran número de falsas correlaciones (falsos positivos), por dos razones fundamentales: que la lógica de predicados no haya sido correctamente configurada o que la calidad de los sensores de la instrumentación de es la adecuada.

La correlación basada en secuencias o correlación secuencial puede ser dividida en varias categorías, dependiendo de la forma de representación de los distintos escenarios (Salah, y otros, 2013):

- Pre condiciones.
- Post condiciones.
- Grafos.
- Modelos de Markov.
- Redes Bayesianas.
- Redes Neuronales entre otras técnicas.

1.2.3 Modelos basados en casos

Los modelos de correlación basados en casos defienden la idea de la existencia de un sistema basado en conocimiento usado para representar escenarios bien definidos. En este sentido han sido desarrollado una gran cantidad de métodos de minería de datos para patrones específicos, así como varias técnicas de correlación con el uso de plantillas de escenarios conocidos. Estas plantillas son usadas tanto por los seres humanos usando reglas de expertos o lenguajes de

correlación, como por la inferencia con el uso del aprendizaje por máquinas (*machine learning*) o técnicas de minería de datos. Cuando un problema es resuelto satisfactoriamente, la solución y sus partes son almacenas en una base de conocimientos, conocida como base de casos, es en ese entonces cuando se desea analizar un nuevo caso, el sistema revisa en la base de casos y encuentra el caso más similar teniendo en cuenta fundamentalmente los atributos claves, y cuales atributos serán usados para indexar y acceder a un caso.

Para responder estas preguntas varios algoritmos han sido desarrollados e implementados como: el vecino más cercano, Inductivo y la indexación basada en conocimiento. La correlación basada en casos en eficiente para resolver problemas muy conocidos especificando un plan de acción completo o en escenarios previamente observados. Por tanto, estas ventajas permiten a los expertos descubrir nuevos escenarios y sus posibles soluciones. Sin embargo, en algunos casos nos es fácil conocer todos los posibles escenarios y crear además una base con un grupo de casos que describa adecuada y correctamente las posibles soluciones. Además, la eficiencia en cuanto al tiempo podría ser poco viable al calcular las relaciones de eventos en tiempo real.

Las soluciones existentes se pueden agrupar en dos categorías fundamentales: basado en expertos y conocimiento inferido.

En la categoría de **modelos basados en el conocimiento de expertos**, la base de datos de conocimiento es construida mediante la intervención de los seres humanos, o expertos. Este conocimiento se formula tanto usando reglas de expertos como escenarios predefinidos, tendiendo a imitar el conocimiento de un ser humano, el cual puede estar dado como resultado de la experiencia o del entendimiento del funcionamiento de un sistema. Una de las principales desventajas de este tipo de técnicas es la escalabilidad, debido a la frecuente actualización y evolución de los sistemas.

En el caso de los modelos basados en la **inferencia de conocimiento**, son usados generalmente implementando métodos de inferencia con algoritmos de aprendizaje por computadoras, en los que las reglas de clasificación son creadas automáticamente a partir de casos de entrenamiento. Las tareas de clasificación pueden ser definidas partiendo de un grupo de ejemplos de entrenamiento, encontrando un grupo de reglas de clasificación que pueden ser usadas para la predicción o clasificación de nuevas instancias. Una de las principales ventajas de estos modelos es la de no asumir a priori el modelo que será usado en el proceso de correlación, esto se hace a partir del proceso de aprendizaje partiendo de un grupo de instancias de entrenamiento (Salah, y otros, 2013). No obstante, es un problema abierto lograr modelos capaces de la generalización de los resultados para eventos no observados en el grupo de entrenamiento. Por otro lado, uno de los grandes inconvenientes lo constituye el alto costo computacional implícito en los proyectos, lo cual asume proporciones mucho más dramáticas en los sistemas de tiempo real.

Se decidió emplear este tipo de modelo de inferencia de conocimientos, por tratarse del SAINUX, que es un sistema que, aunque realiza el procesamiento, la supervisión y el control en tiempo real,

su campo de aplicación actual se enmarca en escenarios de mediana y baja complejidad, en los que las bases de datos históricos no manejan un volumen de datos muy elevados. Además, teniendo en cuenta que los datos almacenados no están fijados necesariamente a un proceso con reglas claramente definidas, sino que estos procesos presentan mayormente una gran heterogeneidad por lo que es necesario crear dinámicamente las reglas para predecir el comportamiento.

1.3 Tipos de conocimiento

De una base de datos se pueden extraer diferentes tipos de conocimiento, que son necesarios identificar y comprender antes de adentrarse en el uso de técnicas que persigan este fin. El conocimiento se puede clasificar según las siguientes categorías (Villena Román, y otros, 2011):

- Conocimiento Evidente: información fácilmente recuperable mediante una simple consulta SQL (Lenguaje de Consulta Estructurado).
- Conocimiento Multidimensional: se trata de interpretar una tabla con **n** atributos independientes como un espacio de **n** dimensiones, lo que permite detectar algunas regularidades difíciles de observar. Este tipo de información es la que analizan las herramientas OLAP (Procesamiento Analítico en Línea).
- Conocimiento oculto: información no evidente, desconocida a priori y potencialmente útil, que puede recuperarse mediante técnicas de MD.
- Conocimiento profundo: información que está almacenada en los datos, pero que resulta imposible de recuperar a menos que se disponga de alguna clave que oriente la búsqueda.

1.4 Descubrimiento de conocimiento en bases de datos

Los tipos de conocimiento son revelados gracias al proceso denominado Descubrimiento de Conocimiento en Bases de Datos (ver Figura 2), KDD por sus siglas en inglés. El KDD es la extracción automatizada de conocimiento previamente desconocido a partir de grandes cantidades de datos. Entre sus metas se encuentran: procesar automáticamente datos crudos, identificar patrones significativos y presentarlos como conocimiento para satisfacer las necesidades del usuario (Vallejos, 2006).

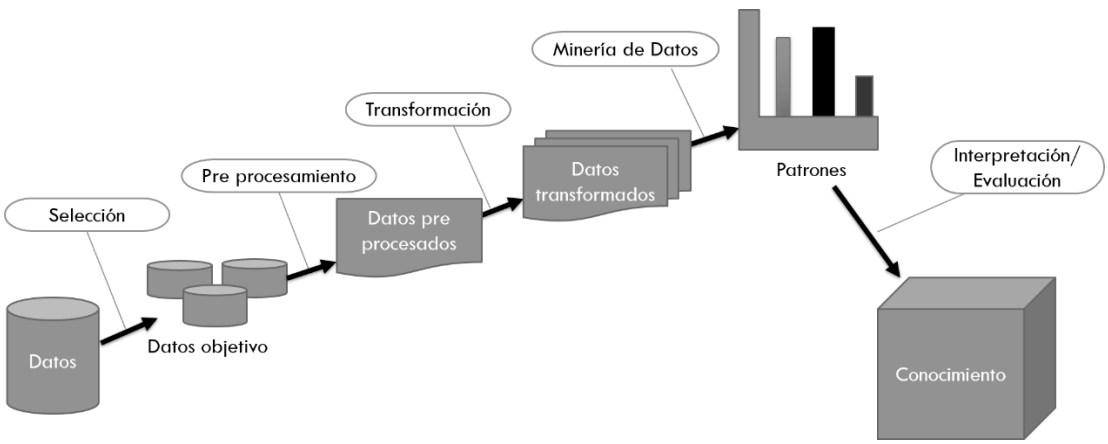


Figura 2: Descubrimiento de Conocimiento en Bases de Datos-KDD. Elaboración propia

Este proceso consta de cinco pasos que se describen a continuación (Cerón Reyes, y otros, 2010):

Paso 1: determinar las fuentes de información, diseñar el esquema de un Almacén de Datos (AD) que consiga unificar toda la información recogida y por último la implantación del AD que permita la navegación y visualización previa de sus datos, para decidir qué aspectos puede interesar que sean estudiados.

Paso 2: selección, limpieza y transformación de los datos que se van a analizar. La selección incluye determinar el conjunto de datos adecuados para el resto del proceso; la limpieza y preprocesamiento de datos se logra diseñando una estrategia adecuada para manejar errores, valores incompletos y anomalías.

Paso 3: seleccionar y aplicar la técnica de MD apropiada. Este paso incluye la selección de la tarea de descubrimiento a realizar y la transformación de los datos al formato requerido por el algoritmo específico de MD.

Paso 4: evaluación, interpretación y representación de los patrones extraídos. Este paso puede involucrar repetir el proceso, quizás con otros datos y otros algoritmos.

Paso 5: difusión y uso. Luego de evaluar los patrones obtenidos es necesario incorporar el conocimiento descubierto al sistema, con el objetivo de realizar acciones o tomar decisiones.

Según varios autores, el proceso de descubrimiento en bases de datos es catalogado como un modelo de procesos, en el que se enuncian las fases necesarias para descubrir conocimiento, pero no se describe cómo cumplir las tareas de minería de datos (Moine, et al., 2010). Por ello se decide estudiar el uso de metodologías para MD, con el objetivo de lograr definir con exactitud tanto las fases del proceso como las tareas a cumplir y la forma de llevarlas a cabo.

1.5 Minería de datos

La MD es la fase del proceso de KDD que consiste en la aplicación de algoritmos que generan una enumeración de patrones. Permite descubrir información valiosa que se encuentra oculta, como asociaciones, anomalías y estructuras significativas a partir de grandes cantidades de datos. La forma tradicional de convertir datos en conocimiento consta del análisis e interpretación que se realiza de forma visual, cual es lento y subjetivo, y además es casi impracticable cuando el volumen de datos adquiere cierto nivel. Según (Hernández Orallo, 2008) este constituye el principal objetivo de la Minería de Datos: resolver problemas analizando los datos presentes en las bases de datos.

Otros autores importantes y de referencia obligada definen la MD como:

- Proceso no trivial para la extracción de información implícita, previamente desconocida, y

potencialmente útil desde los datos (Fayyad, 1996), (Fayyad, 1996a), (Piatetsky, 1991).

- Proceso de extracción de información previamente desconocida, válida y procesable desde grandes bases de datos para luego ser utilizada en la toma de decisiones (Cabena, 1998).
- Análisis de un gran conjunto de datos para encontrar relaciones desconocidas y resumir la información de forma que sean comprensibles y útiles para el usuario de los datos (Hand, 2001).
- Exploración y análisis, a través de medios automáticos y semiautomáticos, de grandes cantidades de datos con el fin de descubrir patrones y reglas significativos (Berry, 1997), (Berry, 2004).

A decir de (Calderón Méndez, 2010), la Minería de datos es un conjunto de herramientas y técnicas de análisis de datos, que por medio de la identificación de patrones extrae información interesante, novedosa y potencialmente útil de grandes bases de datos, que puede ser utilizada como soporte para la toma de decisiones.

En sentido general y teniendo en cuenta las definiciones de muchos de los principales autores sobre el tema, se puede resumir que esta es una disciplina donde se combinan un conjunto de algoritmos y técnicas con el objetivo de extraer conocimiento implícito, previamente desconocido y potencialmente muy útil, que se encuentra almacenado en grandes sistemas de bases de datos, como puede ser la Base de Datos Históricos de un sistema SCADA o del SAINUX específicamente. Por tanto, una de las tareas fundamentales de la MD es descubrir modelos no triviales a partir de los datos y el empleo de estos modelos, debe ayudar a tomar decisiones con un índice de certeza más elevado que reporten beneficios para la organización en cuanto a seguridad, tiempo de respuesta, calidad, eficiencia u otros indicadores.

1.5.1 Ventajas y desventajas de minería de datos

El objetivo de la MD es explorar los datos que se encuentran en las bases de datos, entre las principales **ventajas** que brinda se pueden mencionar (Medina, y otros, 2010):

- Genera modelos descriptivos: posibilita a las empresas explorar y comprender los datos e identificar patrones, relaciones y dependencias que impactan en los resultados finales.
- Genera modelos predictivos: permite predecir valores de variables, y de esta forma apoyar la toma de decisiones de los especialistas de las empresas.
- Permite definir estrategias competitivas: la información obtenida a través de la MD ayuda a los usuarios a elegir cursos de acción y a definir estrategias competitivas, porque conocen información que sólo ellos pueden emplear.
- Cuenta con herramientas que realizan todo el proceso de búsqueda: la búsqueda de conocimiento es realizada por herramientas que automáticamente buscan patrones de comportamiento presentes en los datos.

Las desventajas que tiene la MD están relacionadas con la ética y la seguridad de la información (Álvarez Prados, 2009), por ejemplo:

El uso indebido de la información: las tendencias obtenidas a través de la MD, destinadas a ser utilizadas para fines de mercadotecnia o para algún otro fin ético, pueden ser usurpadas. Las empresas contrarias a la ética pueden utilizar la información que se obtuvo aplicando MD, para aprovecharse de las personas (tanto jurídicas como naturales) vulnerables o discriminar un grupo de individuos.

1.5.2 Técnicas de minería de datos

Las técnicas de MD provienen de la Inteligencia Artificial y de la Estadística. Estas técnicas no son más que algoritmos que se aplican sobre un conjunto de datos con el objetivo de obtener patrones que reflejen el comportamiento de estos. Se clasifican en dos modelos (Riquelme, 2009):

Modelo Predictivo: el sistema encuentra patrones para predecir los valores de una clase, como es el caso de las tareas de clasificación o regresión.

Modelo Descriptivo: el sistema encuentra patrones para presentarlos a un experto en una forma comprensible para él, y que describen y aportan información de interés sobre el problema y el modelo que se obtiene a partir de los datos. Entre las tareas principales con un objetivo descriptivo se encuentran el agrupamiento (*clustering*) y la asociación.

Cada modelo lleva asociado una o varias técnicas de MD, posteriormente se presenta una descripción de las principales técnicas correspondientes a cada uno de estos modelos:

Predictivo

Árboles de decisión: de todos los métodos de aprendizaje, los sistemas basados en árboles de decisión son quizás los más fáciles de utilizar y de entender. Un árbol de decisión es un conjunto de condiciones organizadas, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. El nodo raíz se sitúa en la cima del diagrama del árbol, los atributos son probados en los nodos de decisión, con cada posible indicador del resultado en una rama. Cada rama conduce también a otro nodo de decisión o a un nodo terminal (Ruiz Vera, y otros, 2013).

La principal ventaja de los árboles de decisión es la facilidad de interpretación de los resultados, pues es un modelo de caja blanca o proceso comprensible por usuarios no expertos en MD (Obregón Neira, 2008).

Además, un árbol de decisión representa un conjunto de reglas con formato “si - entonces”, y estas reglas se pueden extraer e interpretar de una manera sencilla, proporcionando un alto grado de comprensión del conocimiento utilizado en la toma de decisiones.

Redes bayesianas: las redes bayesianas son grafos probabilísticos que representan un conjunto de variables aleatorias y sus dependencias condicionales a través de un gráfico acíclico. Una red bayesiana puede representar las relaciones probabilísticas entre enfermedades y síntomas. Dados los síntomas, la red puede ser usada para obtener las probabilidades de la presencia de varias enfermedades. Las técnicas y métodos bayesianos son adecuados para trabajar con incertidumbre. Estos métodos permiten inferir, a partir de los datos, modelos probabilísticos que serán usados para hacer razonamientos. Su gran problema es el coste computacional (Romeu Gullart, y otros, 2010).

Redes Neuronales Artificiales (RNA): las redes neuronales son técnicas analíticas que permiten modelar el proceso de aprendizaje de forma similar al funcionamiento del cerebro humano (Rodríguez Suárez, y otros, 2003). Una red neuronal se compone de unidades llamadas neurona, cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. En general, "las redes neuronales son representaciones de modelos matemáticos donde unidades computacionales son conectadas entre sí por un mecanismo que aprende de la experiencia, es decir de los datos que se han tomado" (Romeu Gullart, y otros, 2010).

Las RNA pueden manejar excepciones y entradas de datos anormales, muy importante para sistemas que tratan un amplio rango de datos. La capacidad de las redes neuronales radica en su habilidad de procesar información en paralelo (esto es, procesar múltiples segmentos de datos simultáneamente), lo cual requiere de mucho tiempo. Como el tiempo es un factor esencial en la actualidad, a menudo deja a las redes neuronales fuera de las soluciones viables a un problema.

Descriptivo

Agrupamiento (Clustering): los datos son agrupados basándose en el principio de maximizar la similitud entre los elementos de un grupo, minimizando la semejanza entre los distintos conjuntos. Se forman grupos tales que los objetos de una misma colección son similares entre sí y al mismo tiempo diferentes a los objetos de otro grupo. Al agrupamiento también se le suele llamar segmentación, los algoritmos de clúster intentan segmentar el conjunto de datos en subgrupos relativamente homogéneos, donde la similitud de los registros dentro de un grupo es maximizada, y la similitud con los registros fuera de este clúster es minimizada (Ruiz Vera, y otros, 2013).

Reglas de asociación: las reglas de asociación tienen como objetivo identificar relaciones no explícitas entre atributos categóricos. Encuentra asociaciones interesantes en forma de relaciones de implicación entre los valores de los atributos de un objeto. En las reglas de asociación se trabaja con dos medidas de calidad, la cobertura y la confianza. La cobertura o soporte es el número de instancias que la regla predice correctamente. La confianza o predicción mide el porcentaje de veces que la regla se cumple cuando se puede aplicar (Romeu Gullart, y otros, 2010).

El análisis realizado sobre las técnicas fundamentales de MD permitió seleccionar las de agrupamiento y reglas de decisión como descriptivos y árbol de decisión para los predictivos. La técnica de agrupamiento es escogida porque tiene como ventaja facilidad de uso y entendimiento del modelo generado, además con la aplicación de esta técnica se obtendrán grupos que reflejen los factores de riesgo que los sistemas para que ocurra un tipo de alarma determinada. Las reglas de asociación se tendrán en cuenta para el descubrimiento de asociaciones y relaciones entre atributos no detectables a simple vista.

Árbol de decisión es seleccionado porque permite determinar un conjunto de condiciones organizadas, posibilitando extraer reglas o patrones presentes en los datos.

En la técnica árbol de decisión el atributo utilizado como clase es “TipoAlarma”, donde cada regla indica la ocurrencia de una alarma. Además, esta técnica, aunque predictiva, se utiliza en la presente investigación con un enfoque descriptivo para determinar relaciones entre los datos. Este enfoque está dado por ser un modelo de caja blanca, permitiendo que los resultados sean fáciles de entender y explicar. Por tanto, debido a la importancia que tiene para los especialistas del área laboral que los resultados sean comprensibles, se llega a la conclusión que estas técnicas se corresponden con las necesidades del proyecto a desarrollar.

1.5.3 Algoritmos de minería de datos

Una técnica constituye el enfoque conceptual para extraer información de los datos y es implementada por varios algoritmos. Cada uno de ellos representa la manera de desarrollar paso a paso una determinada técnica, por tanto, es preciso entender los parámetros y características de los algoritmos para preparar los datos a analizar. Seguidamente se describen los principales algoritmos de MD:

Simple K-Means

Uno de los algoritmos más utilizados para hacer agrupamiento es el Simple K-medias (en inglés *Simple K-Means*) que se caracteriza por su sencillez. En primer lugar, se debe especificar por adelantado cuantos grupos se van a crear, o sea el parámetro k, para el que se seleccionan k elementos aleatoriamente que representarán el centro o media de cada grupo. A continuación, cada una de las instancias o ejemplos es asignada al centro del conjunto más cercano de acuerdo con la distancia (Euclídea, Manhattan, Minkowski, Chebyshev) que le separa de él.

Para cada uno de los grupos así construidos se calcula el centroide (nuevos centros de cada grupo) de todas sus instancias y se repite el proceso completo con los nuevos centros. La iteración continúa hasta que se asignen los mismos ejemplos a los mismos grupos, esto significa que los puntos centrales de los conjuntos se han estabilizado y permanecerán invariables después de cada iteración. El problema fundamental que presenta este algoritmo es determinar el k ideal (Hernández Orallo, 2008).

A-Priori

A-Priori es un algoritmo de aprendizaje de reglas de asociación muy simple, que permite identificar las posibles correlaciones o interdependencias entre distintas acciones o sucesos, pudiendo reconocer cómo la ocurrencia de un suceso o acción puede inducir o generar la aparición de otros. El funcionamiento del algoritmo A-Priori se basa en la búsqueda de los conjuntos de ítems (reglas) que cumplen con determinado umbral de soporte (Wilford Rivera, y otros, 2009).

J48

J48 es un sistema de aprendizaje supervisado que construye árboles de decisión a partir de un conjunto de ejemplos. Este algoritmo es una implementación libre en java del algoritmo C4.5 (mejora del algoritmo ID3 permitiendo el trabajo con atributos continuos), que utiliza el concepto de entropía de la información para la selección de las variables que mejor clasifiquen a la variable objetivo. Se le amplían funcionalidades tales como permitir la realización del proceso de poda mediante reducedErrorPruning o que las divisiones sean siempre binarias con binarySplits. En cuanto a los tipos de atributos admitidos, estos pueden ser cualitativos, cuantitativos y binarios (Molina López, 2006).

Perceptrón simple capa

Un Perceptrón es un modelo de una neurona. En términos de redes neuronales, un perceptrón calcula la combinación lineal de entradas, luego una función de activación por lo general no lineal, es aplicada a esta combinación lineal para producir una salida. La salida es: $Yj = Fj (\sum Wij\ Xi)$, donde Fj representa a la función de activación y Wij son los pesos. Con el perceptrón simple capa la red neuronal aprende los pesos de los datos (Romeu Gullart, y otros, 2010).

Teniendo en cuenta las técnicas escogidas se dispone a utilizar Simple K-Means y J48 como algoritmos de MD. El algoritmo Simple K-Means fue seleccionado por las ventajas que presenta (Acuna, 2008); (Berzal, 2009), entre las que se encuentran:

Computacionalmente rápido, la velocidad puede ser considerable cuando se trata de grandes volúmenes de datos.

- Puede trabajar bien con valores faltantes (*missing values*).
- Es un algoritmo sencillo.
- Solo es necesario especificar un único parámetro.
- Posibilidad de cambiar los puntos iniciales y obtener resultados diferentes.
- Permite el trabajo con atributos cualitativos, cuantitativos y binarios.
- Trabaja eficientemente con grandes cantidades de datos.

El algoritmo **J48** fue escogido teniendo en cuenta las siguientes ventajas (Sánchez Corales, y otros, 2012):

J48 no es afectado por la introducción de datos que no son altamente significativos en el proceso de aprendizaje.

- Fácil interpretación.
- Velocidad computacional.
- Admite atributos cualitativos, cuantitativos y binarios.

1.6 Metodologías para el desarrollo de proyectos de minería de datos

Las metodologías ayudan a las organizaciones a entender el proceso de descubrimiento de conocimiento y proveen una guía para la planificación y ejecución de los proyectos. A partir del año 2000, surgen tres nuevos modelos que plantean un enfoque sistemático para el desarrollo de proyectos de MD: SEMMA (*Sample, Explore, Modify, Model, Asses* - Muestreo, Exploración, Modificación, Modelado, Valoración), Catalyst conocida como P3TQ (*Product, Place, Price, Time, Quantity* - Precio, Lugar, Producto, Tiempo, Cantidad) y CRISP-DM (*Cross Industry Standard Process for Data Mining*), siendo esta última la más utilizada en la actualidad.

SEMMA, creada por el SAS Institute (*Statistical Analysis Systems* - Sistemas de Análisis Estadístico), se define como "el proceso de selección, exploración y modelado de grandes volúmenes de datos para descubrir patrones de negocio desconocidos" (Moine, y otros, 2010). La metodología SEMMA se encuentra enfocada especialmente en aspectos técnicos, excluyendo actividades de análisis y la comprensión del problema que se está abordando. Dicha metodología pasa directamente a la manipulación de los datos de la empresa, a la clasificación de variables e inmediatamente al análisis de los datos.

Catalyst, conocida como P3TQ fue propuesta en el año 2003 por Dorian Pyle, personalidad reconocida internacionalmente en el campo de la minería de datos. Catalyst plantea la formulación de dos modelos: el de Negocio y el de Explotación de Información (Moine, y otros, 2010). El Modelo de Negocio (MII), proporciona una guía de pasos para identificar un problema de negocio y los requerimientos reales de la organización. El Modelo de Explotación de Información (MIII), proporciona una guía de pasos para la construcción y ejecución de modelos de MD.

CRISP–DM, en español Procedimiento Industrial Estándar para realizar MD, creada por el grupo de empresas SPSS, NCR y Daimer Chrysler en el año 2000, es actualmente la guía de referencia más utilizada en el desarrollo de proyectos de MD (Moine, y otros, 2010).

1.6.1 Metodología CRISP–DM

CRISP–DM estructura el proceso de MD en seis fases: análisis del problema, análisis de los datos, preparación de los datos, modelado, evaluación y despliegue. Esta metodología constituye la guía de referencia más utilizada en el desarrollo de proyectos de MD porque se enfoca fundamentalmente en aspectos de negocio, entendimiento del problema que se está abordando, y luego pasa al manejo y análisis de los datos.

Otra de las causas por la que esta metodología se ha convertido en una de las más utilizadas es debido a la facilidad de trabajo que brinda, pues cada fase se desglosa en varias tareas generales de segundo nivel, y estas a su vez se proyectan a tareas específicas, posibilitando saber qué hacer en cada momento durante el desarrollo del proyecto. Además, es un proceso iterativo permitiendo regresar de una fase a cualquier otra, en la Figura 3 solo se muestran las relaciones más comunes entre las fases.

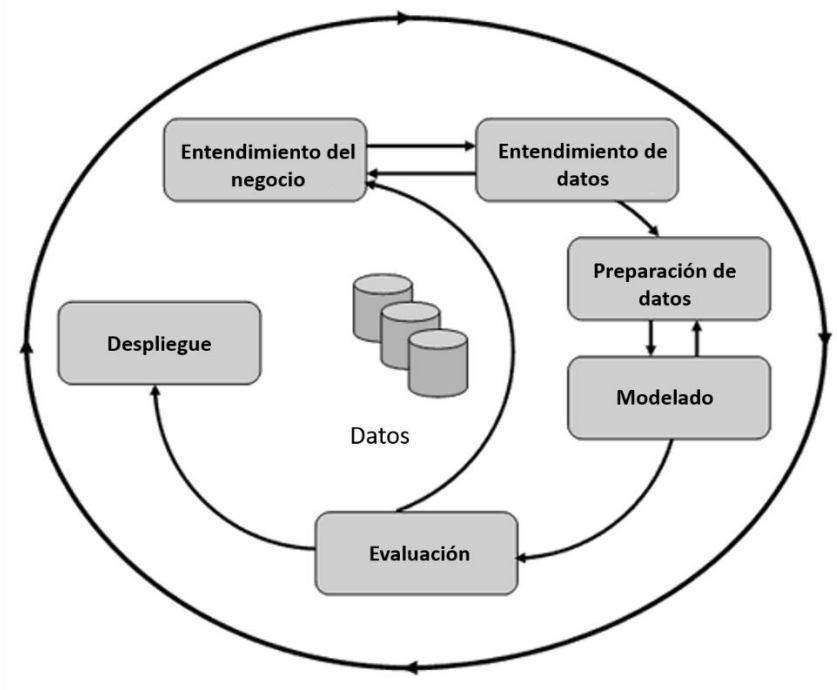


Figura 3: Fases de la metodología CRISP-DM. (Machine Learning Wiki, 2016)

A continuación, se describen las fases de la metodología CRISP-DM (Gallardo Arancibia, 2008):

Análisis del problema

Esta fase se centra en la comprensión de los objetivos del proyecto desde una perspectiva empresarial.

Implica varios pasos claves entre los que se encuentran:

- Establecimiento de los objetivos del negocio: esta es la primera tarea a desarrollar y tiene como metas determinar cuál es el problema que se desea resolver y definir los criterios de éxito del negocio.
- Evaluación de la situación: en esta fase se determinan los requisitos del problema, tanto en términos de negocio como en términos de MD.
- Establecimiento de los objetivos de MD: determinar los objetivos de MD es fundamental, por tanto, si las perspectivas del negocio no pueden ser traducidas de manera efectiva en un objetivo de MD, es prudente redefinir el problema.

Análisis de los datos

Esta fase comienza con una colección de datos inicial. El analista entonces procede a aumentar la familiaridad entre ellos para identificar los problemas de calidad de datos, descubrir ideas iniciales y detectar subconjuntos interesantes que formen hipótesis sobre la información. La fase de análisis de los datos consta de cuatro pasos:

- Recopilación inicial de los datos: esta tarea tiene como objetivo elaborar informes con una lista de los datos adquiridos, su localización y las técnicas utilizadas en su recolección.
- Descripción de los datos: después de adquirir los datos iniciales es necesario hacer una descripción de cada uno de ellos.
- Exploración de los datos: el fin de la exploración es encontrar una estructura general para los datos. Esto involucra acceder a los datos mediante consultas, visualizarlos y la salida de esta tarea es un informe de exploración de los datos.
- Verificación de la calidad de los datos: en esta tarea se efectúan verificaciones sobre los datos, para determinar la cantidad y distribución de los valores nulos y encontrar valores fuera de rango.

La idea en este punto es asegurar la corrección de los datos.

Preparación de los datos

Esta fase cubre todas las actividades para construir el conjunto de datos finales. Las tareas incluyen el registro y selección de atributos, así como la transformación y limpieza de datos. Implica varios pasos claves entre los que se encuentran:

- Selección de los datos: en esta tarea se selecciona un subconjunto de los datos adquiridos en la fase anterior.
- Construcción de los datos: en la construcción se realizan las operaciones de preparación de datos, tales como la generación de nuevos atributos o transformación de valores para atributos existentes.
- Limpieza de datos: esta tarea complementa a la anterior, y es una de las que más tiempo y esfuerzo consume debido a la diversidad de técnicas que pueden aplicarse para optimizar la calidad de los datos. Algunas de las técnicas a utilizar para este propósito son: normalización, discretización de campos numéricos, tratamiento de valores ausentes y reducción del volumen de datos.
- Integración de datos: la integración de datos implica la combinación de información de múltiples tablas o registros.
- Formateo de datos: esta tarea consiste principalmente, en la realización de transformaciones sintácticas de los datos sin modificar su significado, con la idea de permitir o facilitar el empleo de alguna técnica de MD en particular.

Modelado

En esta fase se seleccionan varias técnicas de modelado, algunas de ellas tienen requerimientos específicos sobre la forma de los datos. Por lo tanto, dar un paso atrás a la fase de preparación de los datos puede ser necesario. Los pasos de modelado incluyen la selección de la técnica de modelado y la creación e interpretación de los modelos.

- Selección de la técnica de modelado: esta tarea consiste en la selección de la técnica de MD más apropiada al tipo de problema a resolver. Para esta selección se debe considerar el objetivo principal del proyecto y la relación con las herramientas de MD existentes.
- Creación del modelo: después de seleccionada la técnica, se ejecuta sobre los datos previamente preparados para generar uno o más modelos. Todas las técnicas de modelado tienen un conjunto de parámetros que determinan las características del modelo a generar. La selección de los mejores parámetros es un proceso iterativo y se basa exclusivamente en los resultados generados.
- Interpretación del modelo: en esta tarea los especialistas de MD interpretan los modelos de acuerdo al conocimiento preexistente del dominio y los criterios de éxito preestablecidos. Expertos en el dominio del problema juzgan los modelos dentro del contexto del negocio y expertos en MD aplican sus propios criterios.

Evaluación

Antes de proceder al despliegue final del modelo construido por el analista de datos, es importante evaluarlo más a fondo y revisar su construcción para asegurarse que logre los objetivos del negocio. Es fundamental para determinar si una cuestión importante del negocio no se ha considerado suficientemente. Los pasos claves son:

- Evaluación de los resultados: involucra la evaluación del modelo en relación a los objetivos del negocio.
- Proceso de revisión: se refiere a calificar el proceso de MD, con el objetivo de identificar elementos que pudieran ser mejorados.
- Determinación de los pasos a seguir: si se ha determinado que las fases hasta este momento han generado resultados satisfactorios podría pasarse a la siguiente, en caso contrario es necesario decidirse por otra iteración desde la fase de preparación de los datos con otros parámetros. Podría ser incluso que en esta fase se decida comenzar con un nuevo proyecto de MD.

Despliegue

En general, la creación del modelo no es el fin del proyecto. El conocimiento adquirido debe ser organizado y presentado de una manera que el cliente pueda utilizarlo. Esta fase incluye varios pasos claves, entre los que se encuentran:

- Implementación del plan: esta tarea toma los resultados de la evaluación y concluye una

estrategia para su implementación.

- Monitorización y mantenimiento del plan: si los modelos resultantes del proyecto de MD, son implementados en el dominio del problema como parte de la rutina diaria, es aconsejable preparar estrategias de monitorización y mantenimiento, para determinar si están siendo utilizados apropiadamente.
- Elaboración del informe final: este informe puede ser sólo un resumen de los puntos importantes del proyecto y la experiencia lograda, o puede ser una presentación final que incluya y explique los resultados alcanzados.
- Revisión del proyecto: en este punto se evalúa qué fue lo correcto y qué lo incorrecto, qué es lo que se hizo bien y qué es lo que se quiere mejorar.

En el desarrollo de la investigación se decidió utilizar CRISP-DM en su versión 1.0 como metodología para guiar el proyecto de MD. La selección está sustentada por las siguientes ventajas:

- Es de libre distribución.
- Concibe el proyecto de MD de forma global y estrechamente relacionado al negocio en cuestión.
- Es una metodología que está basada en situaciones reales y actualmente es la guía de referencia más utilizada en el desarrollo de proyectos de MD.
- Fue diseñada de forma neutra a la herramienta que se utilice para desarrollar el proyecto.
- Desglosa cada fase en una serie de tareas posibilitando saber qué hacer en cada momento, además de permitir regresar de una fase a la anterior siempre que sea necesario.

1.7 Herramientas utilizadas

Para extraer conocimiento a partir de los datos, además de contar con una metodología adecuada, es necesario apoyarse en herramientas de software que faciliten la tarea. Seguidamente se darán a conocer las herramientas utilizadas, tanto para realizar el proyecto de MD, como para la preparación de los datos empleados en la generación de los modelos.

1.7.1 Herramientas para el desarrollo de proyectos de minería de datos

Se pueden encontrar tanto en ámbitos comerciales como académicos, una serie de entornos de software diseñados para dar soporte al ejercicio de MD. Posteriormente se muestran algunas de las herramientas de MD más usadas que actualmente están disponibles para el usuario.

WEKA

Weka (*Waikato Environment for Knowledge Analysis* - Entorno para Análisis del Conocimiento de la Universidad de Waikato), es una herramienta visual de libre distribución desarrollada por

miembros de la Universidad de Waikato (Nueva Zelanda). Entre las características de Weka se encuentran (Rodríguez Suárez, y otros, 2003):

- Contiene una extensa colección de técnicas para preprocesamiento de datos y modelado.
- Está disponible libremente bajo licencia GPL (Licencia Pública General).
- Está completamente implementada en Java y puede correr en cualquier plataforma.
- Soporta varias tareas estándar de MD, entre las que se encuentran: preprocesamiento de datos, agrupamiento, clasificación, asociación, visualización y selección.
- Proporciona acceso a base de datos vía SQL gracias a la conexión JDBC (*Java Database Connectvity*) y puede procesar el resultado devuelto por una consulta hecha a la base de datos.
- Entre los formatos que utiliza Weka se encuentran ARFF, CSV, C4.5, JSON y libsvm.
- Se pueden probar diferentes algoritmos de aprendizaje al mismo tiempo y comparar los resultados.

RapidMiner

La herramienta RapidMiner forma parte del proyecto Rapid-i. El proyecto surgió en el 2006 como Spin-Off de la Universidad de Dortmund, donde se inauguró la primera versión del software en el 2001. Entre las características principales de RapidMiner se destacan:

- Está desarrollado en Java.
- Es multiplataforma.
- Es muy sencillo de usar.
- Permite el desarrollo de programas a través de un lenguaje script.
- Incluye gráficos y herramientas de visualización de datos.
- Es un software de código abierto.

Además, esta aplicación ofrece más de 500 operadores para todos los procedimientos de máquina de aprendizaje, y también combina esquemas de aprendizaje y evaluadores de atributos del entorno Weka (García González, 2013).

SAS Enterprise Miner

La herramienta SAS Enterprise Miner es desarrollada por la Empresa SAS (*Statistical Analysis Systems* - Sistemas de Análisis Estadístico). Este software incluye su propia base de datos de información para almacenar y manejar los datos (Rodríguez Suárez, y otros, 2003).

Entre las principales características de esta herramienta se destacan:

- Alta integración con otras bases de datos debido a la gran experiencia de la empresa para operar con grandes volúmenes de datos.
- Proporciona herramientas de modificación y selección de los datos lo que redundará en una mejora de su calidad, en un mejor modelado y en resultados más fiables.

- Es un entorno dinámico e interactivo que está optimizado para visualizar los datos y comprender sus relaciones.
- Ofrece uno de los conjuntos más completos de algoritmos de modelado predictivo y descriptivo.
- No es multiplataforma.
- La licencia no es libre.

En resumen, se trata de una de las herramientas con más potencia en el mercado desde el punto de vista de trabajar con grandes bases de datos, sin embargo, hay que pagar un alto precio por su licencia.

El análisis detallado de las herramientas de MD permitió seleccionar Weka en su versión 3.8.0. Es interesante destacar que teniendo en cuenta que se trata de una herramienta bajo licencia GPL es posible actualizar su código fuente para incorporar nuevas utilidades o modificar las ya existentes. Es multiplataforma y contiene una extensa colección de técnicas para preprocesamiento y modelado de datos. Permite la combinación de varios algoritmos basados en técnicas de MD para obtener mejores resultados en el descubrimiento del conocimiento. Además, se ha seleccionado Weka porque posee características acordes a las necesidades del proyecto y se tiene dominio en el trabajo con la herramienta.

1.7.2 Herramienta para la exploración y transformación de los datos

DataCleaner V1.5.4

DataCleaner es una aplicación Open Source (Código Abierto) para el perfilado, la validación y comparación de datos. Estas actividades ayudan a supervisar y administrar la calidad de los datos, con el fin de garantizar que la información sea útil y aplicable a una situación de negocio. Además, permite obtener una visión de la información, estructura de los datos y reglas de transformación para el proceso de Extracción, Transformación y Carga (ETL) (Pérez Nieblas, y otros, 2014). Para realizar la exploración de los datos y verificar la calidad de estos se utilizará la herramienta Data Cleaner en su versión 1.5.4.

Pentaho Data Integration V5.0.1

Esta herramienta es una de las soluciones más extendidas y mejor valoradas del mercado. Permite realizar transformaciones a los datos de una forma muy sencilla, y es aplicable a diversos tipos de base de datos como son SQL server, PostgreSQL y MySQL. Es de código abierto y sin costes de licencia, las características básicas de esta herramienta son:

- Entorno gráfico de desarrollo.
- Uso de tecnologías estándar: Java, XML (*Extensible Markup Language* - Lenguaje de Marcas Extensible) y JavaScript.
- Multiplataforma: Windows, Macintosh, Linux.

- Basado en dos tipos de objetos: transformaciones (colección de pasos en un proceso de ETL) y trabajos (colección de transformaciones).

Las soluciones de Pentaho están escritas en java y también tienen un ambiente de implementación basado en java, eso hace que Pentaho sea una solución flexible para cubrir una amplia gama de necesidades empresariales.

Entre sus principales ventajas se encuentran (Vega Torres, y otros, 2008):

- Limpia e integra la información y la pone a disposición del usuario.
- Es utilizada por una gran cantidad de usuarios por ser una de las herramientas libres más antiguas.
- Provee una sola versión de todos los recursos de información, que es uno de los más grandes desafíos para las organizaciones.

Se utilizó la herramienta Pentaho Data Integration para realizar el proceso de ETL, porque aporta los componentes necesarios para obtener los atributos que van a ser utilizados desde la fuente de datos.

Además, permitió obtener los datos en un fichero ARFF, formato utilizado por Weka que es la herramienta seleccionada para desarrollar el proyecto de MD. Un fichero con este formato no sólo contiene los datos desde donde se efectúa el aprendizaje, además incluye meta-información sobre los propios datos como el nombre y tipo de cada atributo (Hernández Orallo, 2008).

1.7.3 Herramientas para el trabajo con la base de datos

PostgreSQL Version 9.4

PostgreSQL es un Sistema Gestor de Base de Datos (SGBD) con su código fuente disponible libremente.

Entre sus principales ventajas se encuentran (Martínez, 2014):

- Diseñado para ambientes de alto volumen.
- Multiplataforma.
- Ahorros considerables en costos de operación.
- Tamaño de base de datos desmedido.
- Es una herramienta libre.
- Soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

Este SGBD permitió la extracción de información y almacenamiento en una base de datos, específicamente de la Base de Datos de Almacenamiento de Históricos del SAINUX desarrollado en CEDIN, además de proporcionar herramientas para añadir, borrar y modificar los datos.

PgAdminIII V1.16.1

PgAdmin es uno de los administradores de bases de datos más utilizados, por sus características de administración de código abierto y porque posee una plataforma de desarrollo para PostgreSQL. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL sencillas hasta el desarrollo de bases de datos complejas. La interfaz gráfica de este administrador es compatible con todas las características de PostgreSQL. Es un software libre publicado bajo la licencia de PostgreSQL. Se utilizó PgAdminIII V1.16.1 como herramienta de administración para la base de datos de PostgreSQL.

1.8 Estándares para el intercambio de información mediante servicios Web

Con el objetivo de integrar la comunicación entre la aplicación desarrollada y los sistemas SCADA, se deciden estudiar las principales tendencias del intercambio de datos entre aplicaciones, los servicios web y sus protocolos más conocidos y establecidos.

XML-RPC

Es un protocolo de llamada a procedimientos remotos que usa Lenguaje de Marcado Extensible (XML por sus siglas en ingles) para la codificación de los mensajes y usa a HTTP como protocolo de transmisión.

XML-RPC es muy simple ya que usa tipos de datos y comandos muy útiles, así como una descripción completa de corta extensión. En comparación con otros protocolos de llamada a procedimientos remotos es muy sencillo, ya que comúnmente este tipo de protocolo consta de una gran cantidad de documentación para sus implementaciones y soporte (Winer, 2001).

SOAP

El protocolo simple de acceso a objetos constituye una evolución del XML-RPC, luego de que el creador de este realizara una asociación con la empresa Microsoft e IBM y decidieran incorporar una serie de funcionalidades. En la actualidad se encuentra bajo el auspicio de la W3C (*World Wide Web Consortium*). Está basada en XML al igual que su predecesor y es ideal para entornos distribuidos. Aun cuando comparte la mayoría de las características de XML-RPC, tiene como principal ventaja sobre este que los parámetros de las llamadas tienen nombre y no es importante su orden, siendo todo lo contrario en XML-RPC (Winer, 2001).

REST

Las siglas REST son el acrónimo en inglés de transferencia de estado representacional. Este es un estilo de arquitectura de software para sistemas hipermedia distribuidos como la red mundial y ha experimentado un gran crecimiento en cuanto a su popularidad desde su propia conceptualización en el año 2000, por el doctor Roy Fielding. Este protocolo posee un conjunto de funcionalidades bien definidas y se basa en el uso de HTTP, presenta determinada escalabilidad y

complejidad en su implementación debido al gran número de objetos que maneja, y es ideal para el intercambio de información con estructuras complejas (Navarro Marset, 2015).

Al realizar este análisis sobre los estándares de intercambio de información mediante servicios web, se arriba a la decisión de usar el protocolo SOAP, debido a las ventajas que presenta respecto a su predecesor XML-RPC, por su sencillez, y la experiencia que presenta el grupo de desarrollo en el uso de esta tecnología.

Conclusiones del capítulo

Teniendo en cuenta el análisis realizado en el capítulo, puede afirmarse que, dentro de los modelos de alertas existentes, el proceso conocido como KDD, con la integración de técnicas de Minería de Datos es una herramienta importante para dar respuestas a preguntas complejas en el negocio de una empresa. Con motivo de que KDD no enuncia cómo cumplir las tareas de MD, se seleccionó la integración de una metodología como CRISP-DM 1.0 para dar solución a esta problemática.

Se analizaron las características del SAINUX, referentes a su modularidad lo cual constituye una ventaja en pos del uso del módulo de Bases de Datos Históricos como fuente de datos, y posibilita la futura integración con el modelo de minería de desarrollado gracias a la flexibilidad en su arquitectura.

Las técnicas de MD seleccionadas fueron árbol de decisión, reglas de asociación y agrupamiento, con el objetivo de analizar y describir el comportamiento de los datos.

Las metodologías utilizadas en el desarrollo de proyectos de MD, exponen un modelo de referencia y una guía para el usuario, con orientaciones y consejos más detallados para el desarrollo de cada fase y tarea.

Las herramientas de MD disponibles permiten automatizar gran parte de la tarea de encontrar los patrones de comportamiento ocultos en los datos. Se seleccionó como herramienta a usar Weka 3.8.0.

Se decidió usar el protocolo SOAP como estándar para el intercambio de información mediante servicios web entre la aplicación desarrollada y el SCADA.

CAPÍTULO 2 PROPUESTA DE SOLUCIÓN

Para lograr el objetivo de realizar un correcto proceso de obtención de patrones con ayuda de la Minería de Datos es necesario atravesar por un conjunto de fases dictadas por la metodología escogida, en este caso CRISP-DM. Realizar un análisis de la Base de Datos Históricos con que se cuenta, y las potencialidades que brinda la estructura de sus tablas. La propuesta contempla el desarrollo de un modelo que contiene los módulos desarrollados, y mediante el uso de la arquitectura modular se logra integrar al SAINUX, haciendo uso además de distintas tecnologías como la Arquitectura Orientada a Servicios (SOA), herramientas de software libre como Weka y Java entre otras.

2.1 Descripción de la solución propuesta

La solución propuesta consiste en la integración de un modelo que contiene módulos que contribuyen al trabajo realizado por los operadores, para el procesamiento de la información contenida en la Base de Datos Históricos. Dicho modelo contempla la creación de un sistema experto encargado de realizar la minería de datos, interactuar con el Sistema de Automatización Industrial y apoyar el proceso de supervisión y control de datos. Además, dicha solución cuenta con una interfaz de comunicación con terceros mediante servicios web, que posibilita el uso mediante el protocolo SOAP de las funcionalidades de la aplicación.

El modelo contempla:

- Extracción del conocimiento implícito en la BDH del SAINUX, aplicando técnicas de MD como agrupamiento y reglas de asociación, empleando los algoritmos K-Means y A-Priori respectivamente.
- Creación de un árbol de decisión mediante el algoritmo C 4.5 (J48 en Weka), para la posterior clasificación de las instancias, y alertas tempranas de alarmas.
- Realización de reportes para la visualización de los patrones descubiertos al aplicar técnicas de MD descriptivas a la información almacenada, mediante reglas de asociación y agrupamiento.
- Incorporación de un módulo de predicción usando J48, capaz de identificar cuando se está en presencia de una situación potencialmente peligrosa.

Figura 4: Descripción de la propuesta de solución. Elaboración propia.

En la propuesta de solución que se presenta se realiza un análisis de los registros de la BDH del SAINUX mediante el **proceso de extracción de conocimiento descrito posteriormente.** Este proceso contempla la fase de MD, en la cual se descubren patrones mediante las técnicas descriptivas de reglas de asociación y agrupamiento, así como la técnica predictiva basándose en árboles de decisión. Como resultado de este proceso se realizan reportes asociados a las técnicas descriptivas y se integra el módulo de predicciones al SAINUX.

2.2 Proceso de descubrimiento de conocimiento en datos BDH-SAINUX

Como se mostró en el epígrafe 1.3, el proceso de descubrimiento de conocimiento en datos consta de varias fases. A continuación, se presenta detalladamente las fases del proceso propuesto teniendo en cuenta además los elementos planteados por el uso de la metodología CRISP-DM.

En un primer instante se muestra el **objetivo** que se persigue al realizar MD a los registros de la BDH de los sistemas industriales. Dichos objetivos constituyen la antesala del proceso **Preparación de Datos**. En este subproceso se realizan varias fases como la selección e integración de los datos de la BDH en un archivo de formato ARFF, típico de Weka.

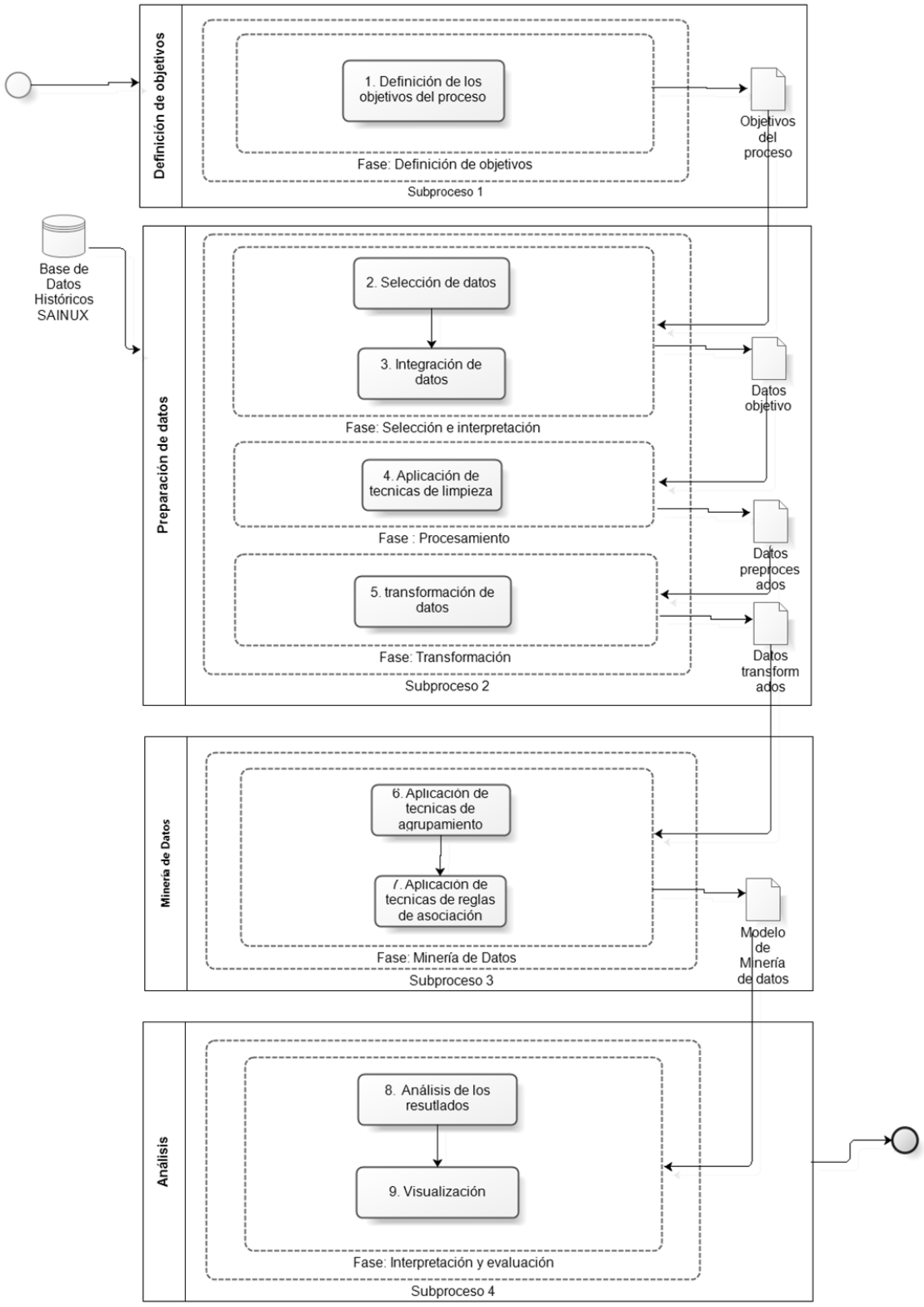


Figura 5: Proceso de descubrimiento de conocimiento en bases de datos (KDD) propuesto.

Luego se realiza el proceso de Minería de Datos donde se aplican las técnicas de minería escogidas, basadas en este caso en algoritmos de clasificación por Árboles de decisión. Obtenido el modelo el sistema se encuentra listo para recibir y devolver datos (mediante el módulo JavaAPP) al interactuar con los sistemas de supervisión industrial interesados en conocer los datos para la prevención de alarmas.

2.3 Subproceso 1- Definición de objetivos

Esta etapa está encaminada a definir los objetivos y requerimientos que se persiguen en consonancia con el negocio, teniendo en cuenta las preguntas propuestas por (Microsoft, 2016):

- ¿Qué se está buscando?
- ¿Qué tipo de relaciones se buscan?
- ¿Se desean realizar predicciones a partir del modelo de Minería de Datos, o simplemente buscar asociaciones y patrones interesantes?
- ¿Qué tipo de datos se tienen y qué información hay en cada columna?

Las actividades realizadas en este subproceso se muestran en la siguiente figura y quedan descritos a continuación.

Fase: Definición de objetivos

Una vez entendidos los objetivos del negocio, se definen los objetivos del proceso de MD dentro del proyecto en términos técnicos y se obtienen como resultados los Objetivos del proceso.

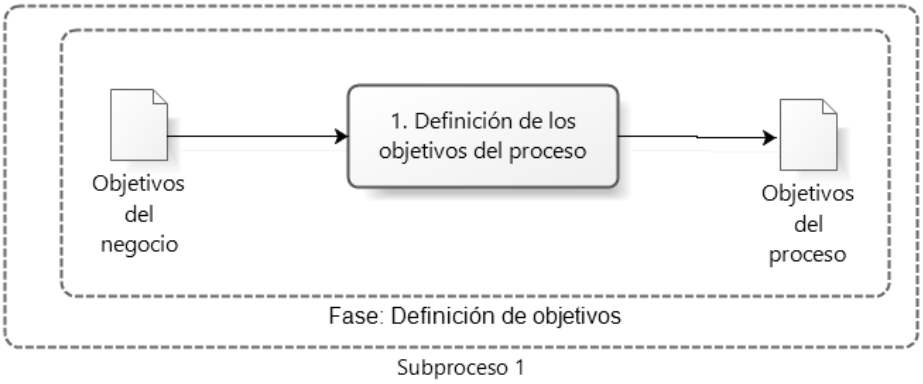


Figura 6: Subproceso 1- KDD

Se tiene como objetivo descubrir el conocimiento implícito en los datos almacenados en los registros de la Base de Datos Históricos del SAINUX a través de la aplicación de tareas de MD como agrupamiento y reglas de asociación. Construir un informe estadístico de los resultados obtenidos al descubrir patrones en el proceso de detección de alarmas y eventos críticos entre otros patrones de comportamiento. Este conocimiento extraído puede revestir una importante utilidad para los operadores y debe ser expuesto de una forma clara y de fácil comprensión para cualquier tipo de usuario relacionado con el control automático.

Para el desarrollo del proceso se seleccionó la herramienta Weka como base para la aplicación de los algoritmos de MD la cual está desarrollada en Java por lo que se tiene como requerimiento la instalación de la máquina virtual de Java. Ambas tecnologías son de libres y multiplataforma por lo que no existen restricciones en este sentido.

Con la definición de estos objetivos queda cumplida la tarea 1, de Definición de los objetivos del proceso de descubrimiento de conocimiento.

2.4 Subproceso 2- Preparación de datos

El propósito fundamental de la fase de Preparación de Datos es el de transformar los datos en crudo de la fuente de datos anteriormente definida (Base de Datos Históricos-SAINUX), de modo que la información contenida en el conjunto de datos pueda ser accesible o accesible de una forma más adecuada.

Este subproceso se enfoca en la selección de los datos fuente, identificar problemas de calidad, balanceo, potencialidades para la extracción del conocimiento luego del procesamiento. Las tareas realizadas en este subproceso se muestran en la Figura 9.

Fase: Selección e integración

Luego del Proceso 1, en el cual fueron determinados los objetivos que se persiguen con el proceso de descubrimiento del conocimiento, se deben seleccionar los datos a los cuales aplicar Minería de Datos. Estos serán la entrada del proceso de descubrimiento y por tanto constituyen la primera etapa de la Selección. Etapa que tiene como objetivos la modelación de las fuentes de datos y su selección.

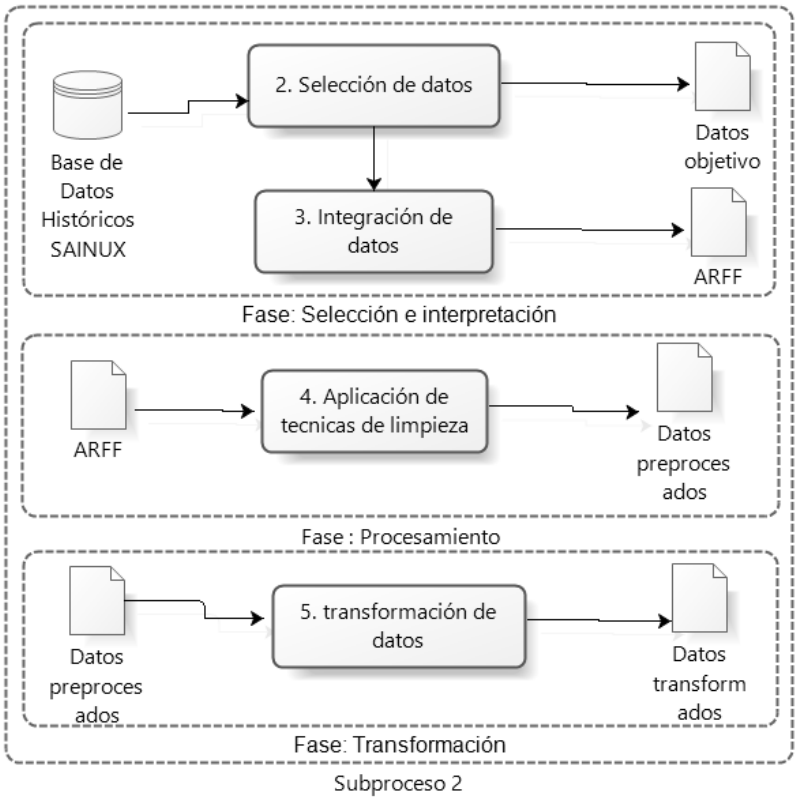


Figura 7: Subproceso 2- KDD

Actividad 2: Selección de datos clave

La base de datos encargada de archivar la información histórica contiene un esquema nombrado '*data*', la cual consta de 10 tablas como se muestra en la siguiente figura:

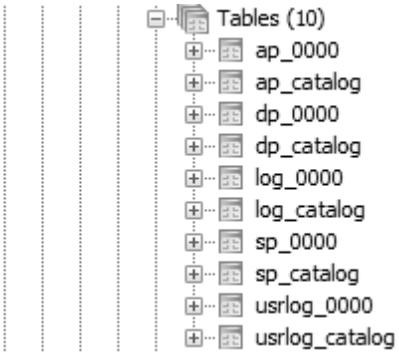


Figura 8: Tablas almacenadas por BD-históricos SAINUX

Las tablas con prefijo 'ap_' contienen información asociada con los puntos analógicos (*analogic point*), 'dp_' a puntos digitales (*digital point*), 'sp_'estado del punto (*status point*), 'log_' almacena los logs o información sobre el funcionamiento de todo el sistema, y finalmente 'urslog_' almacena información sobre los usuarios. Las tablas con sufijo 'catalog' se refieren al catálogo de puntos analógicos con los que se cuenta, de usuarios, puntos digitales, estado de los puntos y *logs* del sistema, en dependencia del prefijo sobre el que actúe.

De las bases de datos históricas con que cuenta el sistema se almacena información de los siguientes elementos, como se muestra en la tabla 1:

Tabla 1: Atributos comunes almacenados en la BD-históricos SAINUX

Elemento	Descripción
Timestamp	Estampa de tiempo, guarda valores asociados con el momento en que ocurre el evento del cual se guarda información
Resource_id	Almacena el identificador del recurso implicado.
User_id	Almacena el identificador del usuario implicado.
Group	Almacena el grupo
Descripcion	Almacena información del tipo de evento al que hace referencia. Para el caso de las alarmas asume valores tales como: • alarma: alto-alto • alarma: alto • alarma: bajo • alarma: bajo-bajo
Previous_value	Almacena el valor que contenía la medición asociado al punto en la medición anterior.
Current_value	Almacena el valor actual asociado al punto.

En el caso del atributo descripción, por su importancia este realiza funciones de clase objetivo (o clasificador), las alarmas según las normas de estándares industriales asumen este tipo de datos para describir su severidad, de modo que “alarma: alto-alto” se refiere a una situación de severidad muy alta, mientas “alarma: bajo-bajo” a una con severidad muy leve.

Se decide ejecutar el algoritmo *CorrelationAttributeEval.Ranker* (evaluación de correlación de atributos) con el objetivo de evaluar la incidencia de los distintos atributos en el proceso de clasificación arrojando los resultados siguientes:

Ranked attributes:

0.34727 1 day_seconds
0.01467 2 resource_id
0.01371 4 current_value
0.00906 5 variation_value
0.00309 3 previous_value
-0.00020 7 block_name
-0.00064 8 group
-0.00081 6 user_id

Al obtener este resultado se deciden obviar atributos con una incidencia con valor negativo, debido al pobre aporte que realizan en los procesos de clasificación y agrupamiento de instancias (Ian, y otros, 2005), de modo que los atributos *day_seconds*, *resource_id*, *current_value*, *variation_value* y *previous_value* son elegidos como atributos claves, y continuarán formando parte del proceso de descubrimiento de conocimiento.

Actividad 3: Integración de datos

Los datos seleccionados se encuentran en una Base de Datos relacional, distribuidas en 10 tablas, como se mostró anteriormente. Por lo que gran parte de la información a extraer se encuentra en las relaciones entre las tablas, lo cual entorpece el procesamiento de los datos a través de Weka. Por tanto, se hace necesario extraer los datos y ordenarlos en una estructura con la que la herramienta Weka pueda interactuar. Esta herramienta permite varias formas para definir el origen de los datos a procesar, los cuales pueden proceder de una Base de Datos NoSQL (no relacional), ficheros binarios, archivos de valores separados por coma (o CSV por sus siglas en inglés) o archivos con formatos ARFF (formato de archivos de atributos-relación por sus siglas en inglés). Para el desarrollo de esta investigación el método para almacenar la fuente de datos es el ARFF, nativo de Weka.

Este tipo de archivos está compuesto por tres partes fundamentales: cabecera, declaración de atributos y datos.

En la cabecera como su nombre lo indica se define el nombre de la relación, siguiendo la siguiente estructura:

```
@relation <nombre_de_la_relación>
```

En la región de declaración de atributos se define el nombre de los atributos que contiene el archivo, seguido del tipo de datos del atributo, siguiendo la estructura:

```
@attribute <nombre_atributo> <Tipo_atributo> o <rango>
```

Los tipos de datos que pueden tener los atributos definidos por Weka son: String, Numeric, Nominal, Integer y Date:

String: Para expresar cadenas de texto.

Nominal: Para expresar los posibles valores que puede alcanzar una variable. Se enuncian entre llaves y separados por coma.

Numeric: Para expresar los números reales.

Integer: Para expresar los datos que representan números enteros.

Date: Expresa fechas, para ello debe ir precedido de una etiqueta de formato entrecomillada la cual está compuesta por caracteres separadores y unidades de tiempo: dd,MM,yyyy,HH,mm,ss.

Finalmente, en la última sección se declaran las instancias de la relación, precedido por la cadena @data, los atributos de la instancia se delimitan por coma (los valores numéricos no enteros utilizan punto en vez de coma), y las instancias se delimitan por saltos de línea.

En la siguiente figura se muestra el fragmento inicial de un fichero con extensión ARFF, el cual muestra el comportamiento de 10 días de recolección del comportamiento de las alarmas en el SAINUX (la cual archiva 431996 instancias en el transcurso de ese tiempo), siguiendo la estructura propuesta por Weka para este tipo de archivos. Se debe señalar que los datos de los cuales se hace uso son resultado del proceso de pruebas realizado al SAINUX en un ambiente controlado, ya que esta solución actualmente se encuentra en fase de liberación y no se está en explotación por empresas a nivel nacional.

SAINUX_data_mining_model.arff

```
@RELATION SAINUX_data_mining_model

@ATTRIBUTE day_seconds real
@ATTRIBUTE resource_id real
@ATTRIBUTE previous_value real
@ATTRIBUTE current_value real
@ATTRIBUTE variation_value real
@ATTRIBUTE description {alarma:bajo-bajo,alarma:bajo,alarma:alto,alarma:alto-alto}

@DATA
1,0,815,797,-18,alarma:bajo
2,0,158,766,608,alarma:alto-alto
3,0,330,837,507,alarma:alto-alto
5,0,67,163,96,alarma:bajo
7,0,162,332,170,alarma:alto-alto
8,0,217,743,526,alarma:alto-alto
9,0,813,537,-276,alarma:alto-alto
10,0,727,711,-16,alarma:bajo
13,0,344,410,66,alarma:bajo
17,0,592,914,322,alarma:alto-alto
18,0,779,448,-331,alarma:alto-alto
21,0,330,468,138,alarma:alto-alto
24,0,680,911,231,alarma:alto-alto
26,0,392,902,510,alarma:alto-alto
30,0,363,268,-95,alarma:bajo
33,0,826,820,-6,alarma:bajo
34,0,431,719,288,alarma:alto-alto
35,0,66,265,199,alarma:alto-alto
36,0,717,510,-207,alarma:alto-alto
38,0,852,249,-603,alarma:alto-alto
```

Figura 9: Ejemplo del archivo SAINUX_data_mining_model_14_2_016_to_24_2_016.arff

El resultado de esta fase (un archivo con formato ARFF), constituye la antesala para la siguiente fase, en este caso la fase de Pre-procesamiento.

Fase Pre-procesamiento

El objetivo de esta fase es el de contribuir a garantizar la calidad de la información sobre la que se espera extraer el conocimiento antes de avanzar a la aplicación de técnicas de Minería de Datos. La calidad de la ejecución de esta fase incide directamente en la calidad de los modelos generados a partir de dicha información y su posterior utilización en la toma de decisiones. Es evidente que la obtención de información útil para ser procesada posteriormente es un factor clave en el descubrimiento de conocimiento (Rodríguez, 2014).

Llegado el momento de crear el archivo con extensión ARFF, es muy importante revisar el nivel de la calidad de los datos que lo conforman, de no ser así el trabajo inadecuado con la información y la unión innecesaria de datos de distintas fuentes puede generar anomalías en los resultados del análisis de los datos. Estas anomalías pueden provocar inferencias erróneas sobre los datos almacenados y por consiguiente que se tomen decisiones incorrectas, lo cual es un riesgo que debe ser mitigado, dada la importancia y criticidad de los procesos que manejan regularmente los Sistemas de Automatización Industrial, con consecuencias devastadoras para las vidas humana, la economía y el patrimonio en general. Para evitar estos problemas se aplican técnicas de limpieza de datos como parte de la etapa de Pre-procesamiento (Montes de Oca, 2015).

Actividad 4: Aplicación de técnicas de limpieza

El objetivo de esta actividad consiste en preparar y simplificar el problema en cuestión, cuidando de no excluir o dañar datos de importancia para el proceso de modelado. Uno de los elementos es el relacionado con el rendimiento de la aplicación, al reducir el volumen de datos de entrada que se deben analizar. Otra de las operaciones que se realizan en esta actividad es el trabajo con los valores ausentes, y la eliminación o tratamiento (según se decida) de los datos incorrectos. Respecto a estos dos últimos casos es importante destacar que teniendo en cuenta el negocio del cual surge la fuente de datos (en este caso el SAINUX), los datos tanto omitidos como incorrectos, forman parte del funcionamiento normal del proceso de supervisión y control industrial, y generalmente traen aparejada una respuesta por parte del sistema consistente en alarmas, eventos críticos o simplemente advertencias, lo cual puede formar parte importante del procesamiento que se desea realizar.

Por otro lado, no deja de ser importante analizar la existencia de valores erróneos como caracteres extraños, incoherencias notables y otros aspectos que, si bien no son del dominio de los usuarios no relacionados con estos sistemas, si lo son por parte del personal capacitado como operadores de centros de control, especialistas funcionales en automática y demás.

Como solución a esta fase se implementó una aplicación desarrollada en Java, la cual carga el archivo con extensión ARFF con los datos, analiza el contenido de cada una de las instancias y toma decisiones por cada una de ellas siguiendo las reglas del siguiente esquema:

Tabla 2: Esquema de decisiones en técnicas de limpieza

Atributo	Problemas	Decisión
Timestamp	Valores ausentes	Eliminar instancias
	Valores incoherentes (fechas anteriores al año 2000, o posteriores a la fecha actual)	Eliminar instancias
	Formato incorrecto (caracteres extraños)	Eliminar instancias
	Formato incorrecto (formato de estampa de tiempo incoherente tal como fechas en formato mm/dd/aaaa y otros).	Intentar convertir los datos a estampa de tiempo legible, si fructifica se deben Actualizar las instancias implicadas, de lo contrario Eliminar instancias
Resource_id	Valores ausentes	Eliminar instancias
	Formato incorrecto (caracteres extraños)	Eliminar instancias
Descripcion	Valores ausentes	Eliminar instancias
	Formato incorrecto (caracteres	Eliminar instancias

	extraños)	
	Formato incorrecto (valores diferentes a los nominales esperados como “alto-alto”, “alto”, “bajo”, “bajo-bajo”)	Eliminar instancias
Previous_value	Valores ausentes	Eliminar instancias
	Formato incorrecto (caracteres extraños)	Eliminar instancias
Current_value	Valores ausentes	Eliminar instancias
	Formato incorrecto (caracteres extraños)	Eliminar instancias

Al finalizar este proceso la aplicación renombra el archivo original, poniendo como sufijo del nombre una secuencia consistente en el año, día, mes, hora, minuto, segundo y milisegundos para lograr la identificación de forma unívoca, por ejemplo, el archivo con nombre: “SAINUX_data_mining_model_14_2_016_to_24_2_016.arff” pasaría a llamarse “SAINUX_data_mining_model_14_2_016_to_24_2_016_20160204_101056_624.arff”.
Posteriormente la aplicación crea un archivo con el nombre original, en este caso “SAINUX_data_mining_model_14_2_016_to_24_2_016.arff”, el cual contiene los datos originales con las actualizaciones luego de ejecutar la actividad de limpieza de datos, culminando así con la fase de Pre-procesado.

Fase: Transformación

Esta fase se encuentra estrechamente relacionada con la fase anterior, a causa de que para transformar los datos y aplicar técnicas del subproceso de Minería de Datos, estos deben pasar necesariamente por un pre-procesado. La actividad que se lleva a cabo en esta etapa es la de transformación de los datos.

Actividad 5: Transformación de datos

Para asegurar un mejor proceso de extracción de conocimientos, se decide modificar la naturaleza de determinadas variables, desechando atributos que no serán de ayuda, creando otros que sí lo serán y modificando otros, como se muestra en la tabla 3:

Tabla 3: Transformación de datos del archivo original

Acción	Atributo	Observaciones
Eliminar	User_id	Este atributo no presenta relevancia para el proceso de descubrimiento de conocimiento. Pues almacena información del usuario asociado a esta operación en caso de existir, lo

		cual es nulo en la mayoría de los casos.
	Group	Este atributo no presenta relevancia para el proceso de descubrimiento de conocimiento, al igual que el atributo aterior se encuentra nulo en la mayoría de los casos.
Modificar	Timestamp	Por la naturaleza del negocio en cuestión se hace importante definir/controlar el comportamiento de los eventos críticos en un momento definido del día, independientemente de la fecha en que ocurra, por lo que se decide crear el atributo “day_seconds” que almacena el segundo del día en que se registró la instancia asociada. Esta decisión está basada en el concepto de técnicas de correlación de alertas basadas en series temporales, específicamente definiendo como ventana de tiempo al transcurso del día iniciado en el segundo 0, y terminando en el segundo 84399, buenas prácticas llevadas a cabo y sugeridas por (Jakobson, y otros, 1995).
Crear	Variation_value	Se decide crear este atributo por la importancia que implica en la ocurrencia de alarmas y eventos críticos en los ambientes de supervisión y control industrial, resultado de la combinación de los atributos “Current_value” y “Previous_value”.

Es importante señalar que en el caso de la primera columna asociada al momento del día en que ocurre la alarma, se desechó la posibilidad de trabajar con el formato de fechas (tipo *Date*), ya que desde el punto de vista de las técnicas de minería de datos más habituales solo interesa distinguir entre dos tipos: NUMÉRICOS y categóricos o DISCRETOS. Por los que al formato de hora se le aplica una conversión del formato HH:mm:ss a entero.

Ejemplo:

`HH:mm:ss = valor` *`Integer`*

`00:30:10 = 1810`

Si bien es cierto que existen varios tipos de datos aceptados por Weka como se mencionó anteriormente (enteros, reales, nominales, fechas, cadenas y otros), pero desde el punto de vista de las técnicas de Minería de Datos más comunes es importante transformar a dos tipos de datos fundamentales: numéricos y categóricos o discretos, esto motivado principalmente porque la

implementación y características intrínsecas de muchos de los algoritmos y técnicas más usados trabajan con estos tipos de datos de una forma mucho más eficiente.

Por ejemplo, algoritmos de clasificación como Red de Bayes (o ***BayesNet*** del inglés) o el de reglas de asociación A-Priori, necesitan y trabajan con atributos discretizados en el caso de los numéricos, ya que no es posible para este algoritmo trabajar con este tipo de datos. Por otro lado, técnicas de asociación como A-Priori también deben realizar discretización a los atributos numéricos.

En orden a permitir en un futuro la clasificación con la ayuda de algoritmos que necesitan datos con atributos discretizados, se decide crear un archivo ARFF con estas características a partir del archivo "SAINUX_data_mining_model_14_2_016_to_24_2_016.arff" antes mencionado y archivarlo en función de ejecutar clasificaciones de prueba. Para ello se ejecuta el filtro:

```
weka.filters.unsupervised.attribute.Discretize
```

Este proceso consiste en transformar atributos numéricos en simbólicos, obteniendo como resultado una serie de etiquetas al dividir la amplitud numérica total del atributo en intervalos (Amaya Torrado, 2014). Si bien el proceso de discretización de variables se puede realizar de manera arbitraria visualizando los valores de esta variable, también se han desarrollado investigaciones que realizan estos trabajos de forma automática como los de (Hu, 2009) y (Hua, 2009).

Al ejecutar el filtro de discretización con las opciones de configuración como se muestran en la Figura 12 se obtiene un resultado similar al mostrado en la Figura 13:

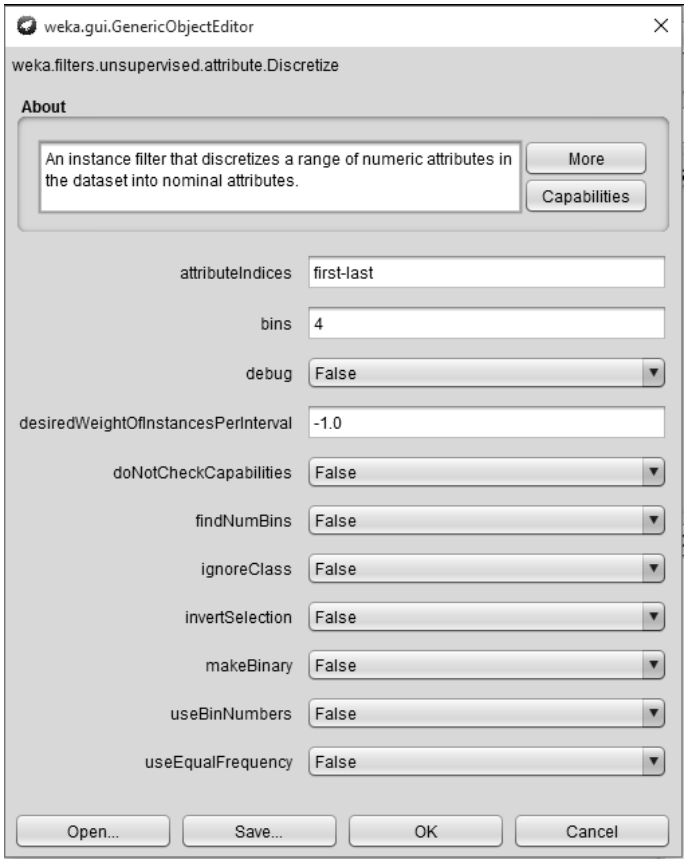


Figura 10: Configuración para ejecución de filtro no supervisado mediante discretización

Nótese que en la configuración escogida se modificó el parámetro *bins* al valor 4, esta variable por defecto engloba al valor 10. Su función es definir la cantidad de intervalos en los cuales serán divididas las amplitudes de las variables del archivo.

```
3  @attribute day_seconds {'\'(-inf-21599.75]\'','\'(21599.75-43199.5]\'','\'(43199.5-64799.25]\'','\'(64
4  @attribute resource_id {'\'(-inf-2.75]\'','\'(2.75-5.5]\'','\'(5.5-8.25]\'','\'(8.25-inf)\''}
5  @attribute previous_value {'\'(-inf-249.75]\'','\'(249.75-499.5]\'','\'(499.5-749.25]\'','\'(749.25-in
6  @attribute current_value {'\'(-inf-249.75]\'','\'(249.75-499.5]\'','\'(499.5-749.25]\'','\'(749.25-inf)
7  @attribute variation_value {'\'(-inf--500]\'','\'(-500--1]\'','\'(-1-498]\'','\'(498-inf)\''}
8  @attribute description {alarma:bajo-bajo,alarma:bajo,alarma:alto,alarma:alto-alto}
9
10 @data
11 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(749.25-inf)\'','\'(749.25-inf)\'','\'(-500--1]\'',alarma:ba
12 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(249.75-499.5]\'','\'(249.75-499.5]\'','\'(-1-498]\'',alarma
13 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(749.25-inf)\'','\'(749.25-inf)\'','\'(-1-498]\'',alarma:alt
14 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(499.5-749.25]\'','\'(749.25-inf)\'','\'(-1-498]\'',alarma:a
15 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(499.5-749.25]\'','\'(249.75-499.5]\'','\'(-500--1]\'',alarma
16 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(749.25-inf)\'','\'(249.75-499.5]\'','\'(-500--1]\'',alarma:a
17 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(249.75-499.5]\'','\'(499.5-749.25]\'','\'(-1-498]\'',alarma
18 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(749.25-inf)\'','\'(499.5-749.25]\'','\'(-500--1]\'',alarma:a
19 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(749.25-inf)\'','\'(249.75-499.5]\'','\'(-500--1]\'',alarma:a
20 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(749.25-inf)\'','\'(749.25-inf)\'','\'(-1-498]\'',alarma:baj
21 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(249.75-499.5]\'','\'(749.25-inf)\'','\'(-1-498]\'',alarma:a
22 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(-inf-249.75]\'','\'(-inf-249.75]\'','\'(-500--1]\'',alarma:b
23 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(249.75-499.5]\'','\'(-inf-249.75]\'','\'(-500--1]\'',alarma
24 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(499.5-749.25]\'','\'(499.5-749.25]\'','\'(-500--1]\'',alarma
25 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(-inf-249.75]\'','\'(749.25-inf)\'','\'(498-inf)\'',alarma:a
26 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(-inf-249.75]\'','\'(499.5-749.25]\'','\'(-1-498]\'',alarma:a
27 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(249.75-499.5]\'','\'(499.5-749.25]\'','\'(-1-498]\'',alarma
28 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(-inf-249.75]\'','\'(749.25-inf)\'','\'(498-inf)\'',alarma:a
29 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(749.25-inf)\'','\'(749.25-inf)\'','\'(-500--1]\'',alarma:ba
30 '\'(-inf-21599.75]\'','\'(2.75-5.5]\'','\'(249.75-499.5]\'','\'(249.75-499.5]\'','\'(-1-498]\'',alarma
```

Figura 11: Archivo de formato atributo relación con atributos discretizados

Al término de este proceso se cuenta con los datos listos para ejecutar el proceso de Minería de Datos de la fase con el mismo nombre.

2.5 Subproceso 3- Minería de datos

Fase: Minería de Datos

Esta frase se dispone a aplicar las técnicas de modelado seleccionadas teniendo en cuenta las características de los datos de entrada o la vista minable previamente seleccionada. Las actividades de este subproceso se representan en la siguiente figura, las cuales se ejecutan de forma no secuencial e indistintamente sin dependencias o relaciones entre ellas.

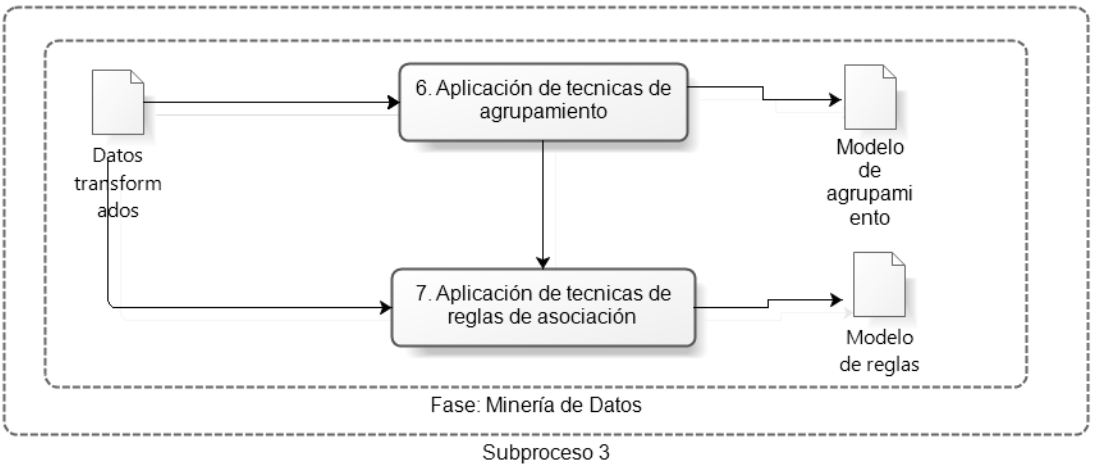


Figura 12: Descripción subproceso 3- KDD

Actividad 6: Técnicas de agrupamiento

Para la actividad de agrupamiento se selecciona el algoritmo K-Means (o vecino más cercano), dentro de la gama de los algoritmos que implementan métodos conocidos como métodos de particionado y recolección. Estos algoritmos de agrupamiento buscan una división del conjunto de datos de entrada en subconjuntos con intersección vacía. Siguen un patrón común que consiste en asignar objetos a los *clústeres* en función de la proximidad de dichos objetos a un representante elegido para cada clúster (Lara, 2011), (Garre, 2007).

Esté método es uno de los más populares en aplicaciones científicas e industriales, su nombre se deriva de la realización del cálculo y representación de cada uno de los clústeres por la media (o media ponderada) de sus puntos, entiéndase por su centroide. El objetivo del algoritmo es dividir el conjunto M de los objetos en un número K de subconjuntos naturales y homogéneos, en los cuales los objetos de cada conjunto sean tan similares entre ellos como sea posible y a la vez lo más distintos posibles del resto de los conjuntos de M (Pérez, 2007).

A continuación, se explican algunos parámetros importantes relacionados con la configuración de la ejecución de este algoritmo:

Función de distancia (***distanceFunction***): Con este parámetro se elige la función de distancia, mediante la cual se calculará el grado de similitud de cada uno de los objetos o instancias en cuestión. Como se puede observar para el caso se emplea el cálculo de la distancia euclídea, la

cual constituye la distancia ordinaria entre dos puntos en un espacio euclídeo, y su cálculo se realiza a partir del conocido teorema de Pitágoras. Téngase los puntos A y B, tal que A y B son puntos formados por dos coordenadas X, Y ((A_x, A_y), (B_x, B_y)), el cálculo de la distancia euclídea sería:

$$d_{A-B} = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2}$$

La fórmula anterior ilustra el modo de calcular la distancia euclídea para un espacio bidimensional (de dos coordenadas), lo cual constituye una variante particular muy conocida de la fórmula que se presenta a continuación, la cual es válida para cuando los puntos A y B se encuentran en un espacio de *n* dimensiones (Martín, 2015):

$$d_{A-B} = \sqrt{\Sigma_1^n (A_n - B_n)^2}$$

Muestreo de desviación estándar (***displayStdDevs***): parámetro binario, usado para mostrar la desviación de los atributos numéricos y la cantidad de atributos nominales. Como su puede observar en la Figura 14, la configuración elegida para este parámetro es verdadero (True).

Reemplazar valores perdidos (***dontReplaceMissingValues***): parámetro binario, usado para elegir si se desea reemplazar los valores perdidos por la media o no. En este caso el uso de este parámetro es innecesario, ya que en una de las etapas anteriores (pre-procesamiento) ya fueron tratados este tipo de datos, por lo que se elige falso (*False*) para la configuración.

Cantidad de Grupos (**numClusters**): número entero, en el que se elige la cantidad de clústeres a generar, en este caso se eligió una cantidad de 5.

Cantidad máxima de iteraciones (**maxIterations**): número entero, con el cual se elige el número máximo de iteraciones a realizar, se configuró este parámetro con el valor de 1000 iteraciones, con el objetivo de aumentar la fiabilidad del método.

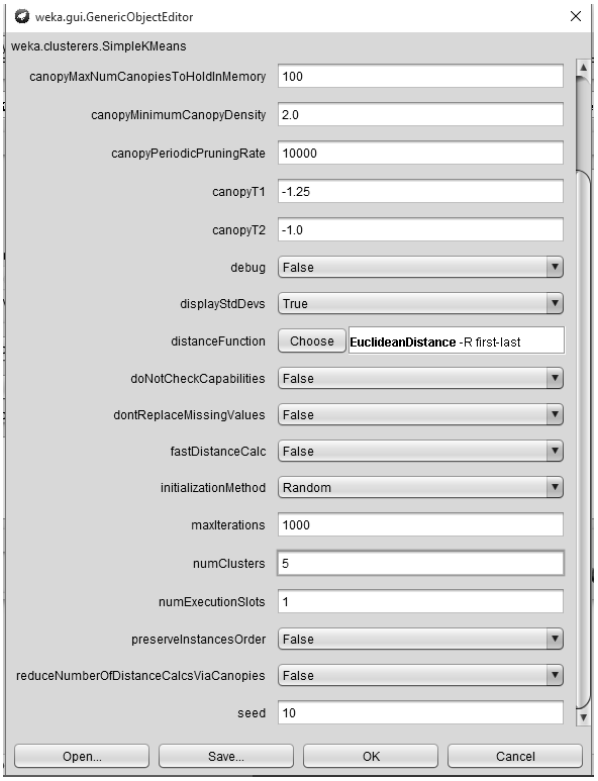


Figura 13: Configuración de parámetros para el algoritmo SimpleKMeans en Weka

Este algoritmo puede ser ejecutado desde la herramienta Weka, luego de cargar el archivo.arff, en la pestaña Cluster, eligiendo la opción de clusterización con el nombre “SimpleKMeans”, eligiendo la configuración de sus parámetros como se muestra en la Figura 14.

Para el resto de los parámetros de la configuración se dejan con los valores cargados por defecto en Weka, el comando a ejecutar en el ambiente Weka en su versión 3.8.0 es el siguiente:

```
weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -V -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 1000 -num-slots 1 -S 10
```

Al ejecutar este comando en la herramienta Weka, sobre el archivo SAINUX_data_mining_model_14_2_016_to_24_2_016.arff, el cual presenta 431996 instancias, se obtienen los resultados mostrados en la tabla 4:

Tabla 4: Instancias por grupos al ejecutar SimpleKMeans

No clúster	Cantidad de instancias	Porciento
0	17726	12 %
1	18562	13 %
2	58858	40 %
3	19447	13 %
4	32286	22 %

```
Final cluster centroids:
                                                Cluster#
Attribute                  Full Data                 0                 1                 2                 3                 4
                           (285117.0)          (34943.0)         (36203.0)         (113837.0)        (37388.0)         (62746.0)
=============================================================================================================================================
day_seconds                43162.3921          56494.2618        55563.0852        25775.7786        53652.6297        53875.9531
                          +/-24930.1767       +/-16150.4956     +/-14736.5691     +/-26635.2541     +/-15291.9112     +/-14514.6094

resource_id                    5.3647              5.2572            8.5978            5.4864            2.1525            5.2524
                              +/-3.4061           +/-3.4072         +/-1.6636         +/-3.2799         +/-1.6775         +/-3.3521

previous_value               499.0218            496.3246          677.5423          498.7033          682.8185          288.5821
                            +/-289.0619         +/-276.3343       +/-226.9374       +/-289.946        +/-224.0914       +/-205.4138

current_value                499.2218            496.9741          323.6014          500.0509          320.1701          706.9884
                            +/-288.6748         +/-276.3038       +/-228.1799       +/-289.5722       +/-227.9729       +/-207.2332

variation_value                0.2                 0.6495          -353.9409           1.3476          -362.6483          418.4063
                            +/-408.4559         +/-57.3745        +/-248.529        +/-417.8587       +/-243.0641       +/-207.2172
```

Figura 14: Modelo obtenido al ejecutar el algoritmo SimpleKMeans con la configuración mostrada

Actividad 7: Técnicas de reglas de asociación

Para el proceso de extracción de reglas de asociación en este caso se utiliza el algoritmo A-Priori, el cual constituye uno de los más utilizados. Este algoritmo en un primer momento construye conjuntos formados por un solo objeto que supere la cobertura mínima, este conjunto de conjunto es usado entonces para construir un conjunto de dos objetos, y así recursivamente hasta que se arribe a un tamaño en el cual no existan conjuntos de objetos con la cobertura requerida (Hernández Orallo, 2008).

Se plantea que todo subconjunto no vacío de un conjunto de objetos frecuentes tiene que ser frecuente también. Si a conjunto de objetos X no satisface el soporte mínimo entonces no es frecuente y si a este conjunto X se le adiciona un objeto entonces el conjunto resultante ocurrirá con menor o igual frecuencia que X (Tanna, 2014).

A continuación se muestran los parámetros de configuración para la ejecución de este algoritmo, es importante señalar que para el uso de este fue necesario cargar el archivo.arff discretizado mencionado anteriormente en la fase de transformación de datos:

Figura 15: Configuración de algoritmo A-Priori en Weka

Algunos de los parámetros más importantes son los siguientes:

car: parámetro binario, que garantiza que se tengan en cuenta todos los atributos para propiciar un mayor número de reglas si se toma en cuenta un solo atributo como clase. En este caso se decide ejecutar con la configuración por defecto (*False*).

classIndex: en este atributo se define con un valor entero el índice del atributo que se va a tomar como clase objetivo, en caso de que el atributo **car** se encuentre en verdadero (*True*).

lowerBoundMinSupport: parámetro que recibe valores reales, y que define el soporte mínimo que deben cumplir las reglas, para el caso se eligió el valor por defecto 0.1.

delta: representa la variación decreciente desde el soporte máximo hasta el mínimo, por cada una de las iteraciones, se mantiene en el valor por defecto 0.05.

metricType: se define qué tipo de métricas se utilizarán para determinar el porcentaje de calidad de las reglas. En este caso también se usa el valor por defecto por métricas de confianza (*Confidence*).

minMetric: define el factor de confianza mínimo que deben cumplir las reglas a obtener. Entre más pequeño este valor más importante las reglas que se obtienen, ya que significa que no se pueden percibir a simple vista. En este caso fue ejecutado el algoritmo con el **minMetric** en 0.2.

numRules: parámetro que recibe un número entero, que representa el número máximo de reglas a obtener, o sea como criterio de parada en caso de que las reglas detectadas sean mayor que ese número que cumplen las restricciones del resto de los parámetros de la configuración. En este caso fueron elegidas un máximo de 20 reglas.

upperBoundMinSopport: en esta variable se determina el soporte máximo desde el cual se comienza a decrecer hasta llegar al soporte mínimo previamente establecido. Este parámetro se definió con el máximo posible (1.0), que es a su vez el valor por defecto.

El resto de los parámetros de configuración se establecieron con el valor por defecto, por lo que la ejecución del comando para el algoritmo de reglas de asociación A-Priori en Weka versión 3.8.0 es el siguiente:

```
weka.associations.Apriori -N 20 -T 0 -C 0.2 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
```

Luego de la ejecución de este comando se obtienen las 20 reglas de asociación deseadas, a partir del conjunto de datos del SAINUX previamente discretizados, como se muestra en la Figura 16. Es importante señalar que para cada una de las reglas se muestran las frecuencias de ocurrencia o cobertura al lado izquierdo y derecho de cada una de ellas, los valores de confianza (conf), elevación (lift), apalancamiento (lev) y convicción (conv). Ambos modelos creados con la ayuda de la herramienta Weka, tanto para las técnicas de agrupamiento como de reglas de asociación constituyen las salida de estas actividades, los cuales son salvados de igual forma en un archivo con formato.arff.

```
Best rules found:

 1. description=alarma:alto 203973 ==> day_seconds='(28799.5-79200.5]' 203973    <conf:(1)> lift:(1.71) lev:(0.2) [84931] conv:(84931.93)
 2. current_value='(73.5-932.5]' description=alarma:alto 172992 ==> day_seconds='(28799.5-79200.5]' 172992    <conf:(1)> lift:(1.71) lev:(0.17) [72031] conv:(72031.81)
 3. day_seconds='(28799.5-79200.5]' 252118 ==> current_value='(73.5-932.5]' 216331    <conf:(0.86)> lift:(1) lev:(-0) [-20] conv:(1)
 4. previous_value='(76.5-906.5]' 358014 ==> current_value='(73.5-932.5]' 307188    <conf:(0.86)> lift:(1) lev:(-0) [-36] conv:(1)
 5. day_seconds='(28799.5-79200.5]' previous_value='(76.5-906.5]' 209088 ==> current_value='(73.5-932.5]' 179354    <conf:(0.86)> lift:(1) lev:(-0) [-71] conv:(1)
 6. description=alarma:alto 203973 ==> current_value='(73.5-932.5]' 172992    <conf:(0.85)> lift:(0.99) lev:(-0) [-2044] conv:(0.93)
 7. day_seconds='(28799.5-79200.5]' description=alarma:alto 203973 ==> current_value='(73.5-932.5]' 172992    <conf:(0.85)> lift:(0.99) lev:(-0) [-2044] conv:(0.93)
 8. description=alarma:alto 203973 ==> day_seconds='(28799.5-79200.5]' current_value='(73.5-932.5]' 172992    <conf:(0.85)> lift:(1.69) lev:(0.16) [70848] conv:(3.29)
 9. day_seconds='(28799.5-79200.5]' 252118 ==> previous_value='(76.5-906.5]' 209088    <conf:(0.83)> lift:(1) lev:(0) [146] conv:(1)
10. day_seconds='(28799.5-79200.5]' current_value='(73.5-932.5]' 216331 ==> previous_value='(76.5-906.5]' 179354    <conf:(0.83)> lift:(1) lev:(0) [71] conv:(1)
11. current_value='(73.5-932.5]' 370711 ==> previous_value='(76.5-906.5]' 307188    <conf:(0.83)> lift:(1) lev:(-0) [-36] conv:(1)
12. day_seconds='(28799.5-79200.5]' 252118 ==> description=alarma:alto 203973    <conf:(0.81)> lift:(1.71) lev:(0.2) [84931] conv:(2.76)
13. day_seconds='(28799.5-79200.5]' current_value='(73.5-932.5]' 216331 ==> description=alarma:alto 172992    <conf:(0.8)> lift:(1.69) lev:(0.16) [70848] conv:(2.63)
14. day_seconds='(28799.5-79200.5]' 252118 ==> previous_value='(76.5-906.5]' current_value='(73.5-932.5]' 179354    <conf:(0.71)> lift:(1) lev:(0) [75] conv:(1)
15. day_seconds='(28799.5-79200.5]' 252118 ==> current_value='(73.5-932.5]' description=alarma:alto 172992    <conf:(0.69)> lift:(1.71) lev:(0.17) [72031] conv:(1.91)
16. previous_value='(76.5-906.5]' 358014 ==> day_seconds='(28799.5-79200.5]' 209088    <conf:(0.58)> lift:(1) lev:(0) [146] conv:(1)
17. previous_value='(76.5-906.5]' current_value='(73.5-932.5]' 307188 ==> day_seconds='(28799.5-79200.5]' 179354    <conf:(0.58)> lift:(1) lev:(0) [75] conv:(1)
18. current_value='(73.5-932.5]' 370711 ==> day_seconds='(28799.5-79200.5]' 216331    <conf:(0.58)> lift:(1) lev:(-0) [-20] conv:(1)
19. previous_value='(76.5-906.5]' 358014 ==> day_seconds='(28799.5-79200.5]' current_value='(73.5-932.5]' 179354    <conf:(0.5)> lift:(1) lev:(0) [71] conv:(1)
20. current_value='(73.5-932.5]' 370711 ==> day_seconds='(28799.5-79200.5]' previous_value='(76.5-906.5]' 179354    <conf:(0.48)> lift:(1) lev:(-0) [-71] conv:(1)
```

Figura 16: Modelo obtenido al ejecutar el algoritmo A-Priori en la herramienta Weka

2.6 Subproceso 4- Interpretación y evaluación

Fase de Análisis:

Al arribar a esta fase se proceden a analizar los modelos obtenidos como resultado de la ejecución de las fases y actividades anteriores, teniendo en cuenta si cumplen con los objetivos

del negocio definidos en la primera fase. La fase de análisis comprende dos actividades fundamentales: el análisis de los resultados obtenidos en las fases anteriores, en este caso grupos, reglas y modelos predictivos y finalmente la actividad de visualización, en la cual los resultados de todo el proceso son brindados al usuario, posibilitando su interacción con estos resultados e integración con el proceso de supervisión y control de procesos a escalas industriales.

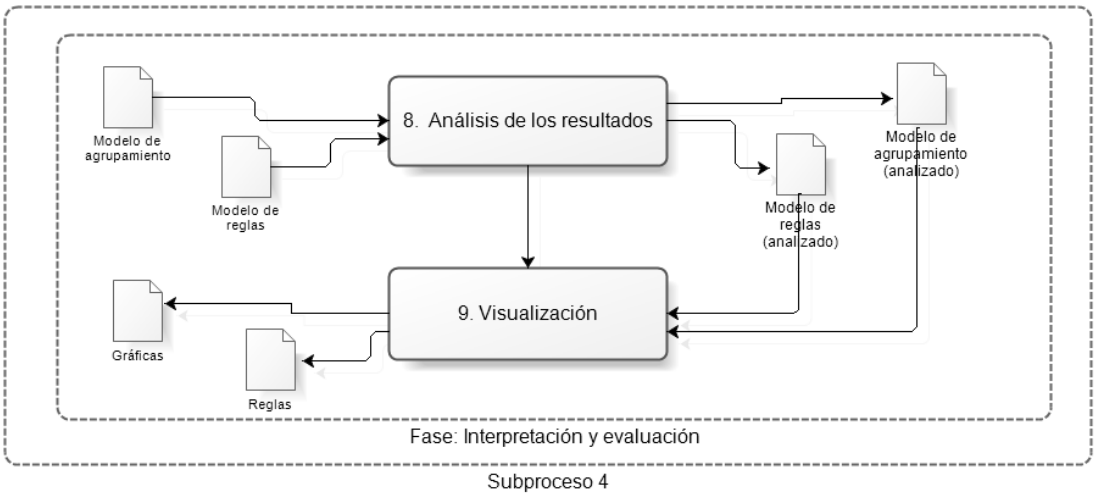


Figura 17: Descripción Subproceso 4- KDD

Actividad 8: Análisis de los resultados obtenidos.

La actividad de medir la calidad de los resultados obtenidos en el proceso de Minería de Datos, dígase patrones descubiertos, asociaciones desconocidas y otros, se convierte en un proceso hasta cierto punto complicado, al no existir una forma viable de ponderar la importancia que revisten estos. El problema se acentúa al tratarse de resultados que comprenden modelos descriptivos.

Al presentarse este tipo de problemática se hace aún más difícil establecer una evaluación coherente de los modelos obtenidos, por lo que se decide contar con la opinión de usuarios eventuales de la aplicación, así como personal especializado en el tema para definir la evaluación de estos modelos. Presentando criterios tales como:

- El interés que despierta en los usuarios los modelos obtenidos.
- El factor de impacto de un modelo respecto al conocimiento previo de los usuarios respecto a determinado problema dentro del negocio a modelar.
- Facilidad de interpretación por parte de los usuarios de los resultados obtenidos.
- Tamaño y complejidad del modelo.
- Capacidad de respuesta al ser usado en ambientes de procesamiento industrial de la vida real.

Estos y otros resultados se verán reflejados en el Capítulo 3 de esta investigación.

Revisión del proceso de descubrimiento de conocimiento en bases de datos

En esta fase del KDD dentro de la actividad 8, de análisis de los resultados obtenidos se procede a realizar una revisión del proceso realizado, atendiendo a que luego de esta actividad se debe proseguir con la actividad de visualización, por lo que asegurar la calidad y correctitud del proceso llegado a esta instancia es muy importante, puesto que el resultado le será presentado a los usuarios finales del sistema, operadores de centros de control, automáticos y directivos para el apoyo en el proceso de toma de decisiones.

En la fase inicial, de comprensión del negocio se centró en entender los objetivos y metas fundamentales del negocio, estableciendo como objetivo fundamental el de descubrir el conocimiento implícito en los datos almacenados en los registros de la Base de Datos Históricos del SAINUX a través de la aplicación de tareas de MD como agrupamiento y reglas de asociación. Construir un informe estadístico de los resultados obtenidos al descubrir patrones en el proceso de detección de alarmas y eventos críticos entre otros patrones de comportamiento.

Posteriormente a esta fase se continuó con la selección de los datos implicados en el proceso por su nivel de relevancia, a partir de la información en crudo almacenada en el módulo de Base de Datos Históricos de los sistemas SCADA, en este caso en el Sistema de Automatización Industrial basado en tecnologías libres.

A continuación, estos datos fueron preparados de modo que pudieran ser interpretados debidamente por las herramientas de Minería de Datos, teniendo en cuenta los estándares de los archivos de atributos y relaciones (ARFF) definidos por la herramienta Weka.

Posteriormente se procede a ejecutar las tareas relacionadas con la aplicación de los algoritmos de MD ya seleccionados teniendo en cuenta la naturaleza de los datos elegidos. Se decide aplicar tareas basadas en agrupamiento, reglas de asociación y predicción de eventos. Específicamente los algoritmos *SimpleKMeans* para el agrupamiento, el algoritmo A-Priori para las reglas de asociación y el algoritmo de árboles conocido como J48 para el proceso predictivo.

Actividad 9: Visualización

La actividad de visualización está encaminada a mostrar de forma gráfica la información almacenada en las bases de datos, de forma amigable, intuitiva y comprensible para el usuario final. Las técnicas de visualización se usan con dos objetivos fundamentales (Hasperué, 2012):

- Aprovechar la capacidad de los seres humanos para detectar patrones en las visualizaciones gráficas, así como anomalías y tendencias facilitando la comprensión de los datos.
- Ayudar al usuario a comprender rápidamente patrones descubiertos automáticamente por un sistema de extracción de conocimiento.

2.7 Diseño de aplicación e integración de la solución con el SAINUX

A modo general la arquitectura de la aplicación propuesta pretende desacoplar la implementación de las aplicaciones brindando la posibilidad de intercambiar un algoritmo de clasificación por otro, en dependencia del negocio que se gestione. Se hace uso de servicios web para la clasificación de un evento bajo determinadas condiciones, específicamente el SCADA es el encargado de realizar consultas a la aplicación de forma periódica previamente definida o al desencadenarse determinadas condiciones. El servicio web se implementa bajo el protocolo SOAP, el cual mediante un procedimiento de llamada y retorno permite invocar un método seleccionado y recibir una respuesta a cambio mediante el protocolo HTTP. El esquema de comunicación usa el patrón cliente-servidor en el que los sistemas de supervisión que acceden a la información de la aplicación pasan a ser clientes del servicio web, el cual funciona como servidor.

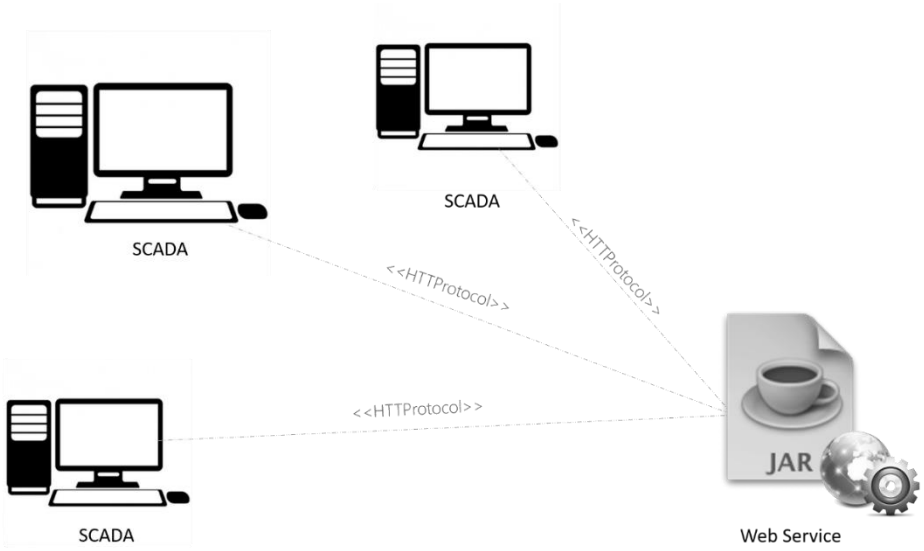


Figura 18: Representación arquitectónica de la interacción sistemas externos- Web Service

Este esquema permite que distintos clientes puedan hacer uso del modelo de minería de datos con el cual se haya compilado la aplicación en java, bajo un ambiente distribuido y siguiendo los estándares SOA.

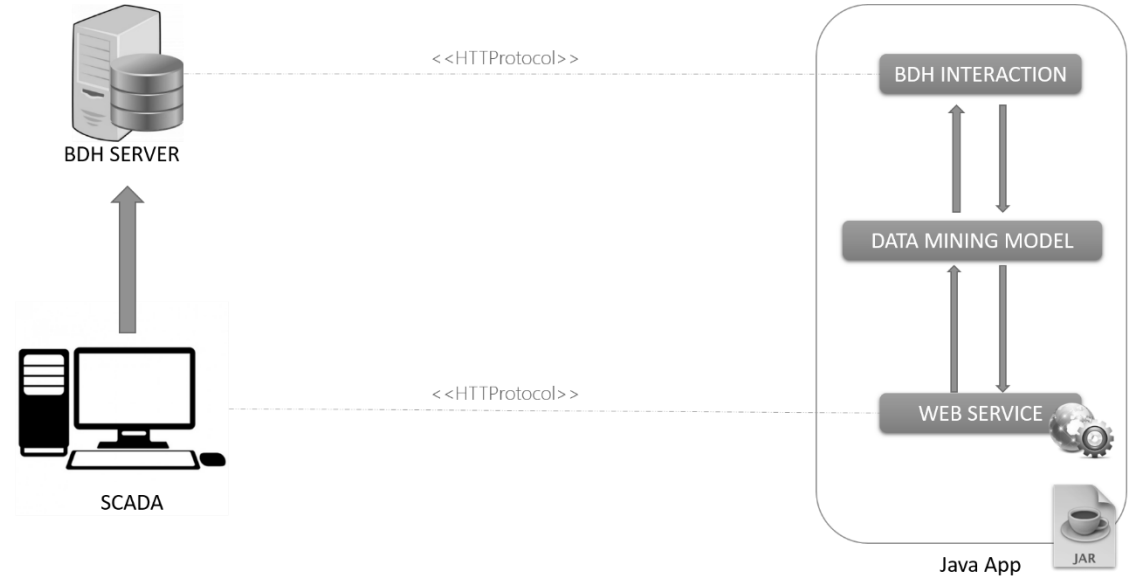


Figura 19: Arquitectura general propuesta

SOA es un estilo de Arquitectura de Software basado en la definición de servicios reutilizables, con interfaces públicas bien definidas, donde los proveedores y consumidores de servicios interactúan en forma desacoplada para realizar los procesos de negocio. Se basa en cuatro abstracciones básicas: servicios, aplicación interfaz (*frontend*), repositorio de servicios y bus de servicios. Un servicio consiste en una implementación que provee lógica de negocio y datos, un contrato de servicio, las restricciones para el consumidor, y una interfaz que expone físicamente la funcionalidad. Las aplicaciones interfaz consumen los servicios formando procesos de negocios. Un repositorio de servicios almacena los contratos de servicios y el bus de servicios interconecta las aplicaciones interfaz y los servicios (DELGADO, 2006). Una de las peculiaridades de esta implementación consiste en el hecho que la aplicación desarrollada en Java crea el modelo de minería de datos dada la dirección de la Base de Datos Históricos del SCADA que realiza la petición.

2.8 Diseño de aplicación en java

Para la arquitectura propuesta se realiza el diseño y desarrollo de una aplicación en lenguaje Java, que tendrá varias responsabilidades. Como se mencionó anteriormente, la elección de este lenguaje para el desarrollo de la aplicación está condicionada por la elección del uso de la API de Weka para el proceso de minería de datos. La aplicación contiene varias clases con sus respectivas funcionalidades como se describe en la imagen 2. Esta aplicación funciona como un cliente de peticiones realizadas por un servidor, en este caso el sistema SCADA, flujo donde la variable “Datos” contiene las condiciones sobre las cuales se desea conocer si son propensas a generar alarmas en el procesamiento del sistema.

La clase Web_service se encarga de invocar a la clase Modelo_mineria_de_datos que es la encargada de la clasificación de estos datos, realizando la carga y el procesamiento de la información almacenada en la Base de Datos Históricos (auxiliándose de la clase Interacción_base_datos) según el estándar de Weka con el formato ARFF (archivo de formato

atributo/relación por sus siglas en inglés), creando un sistema experto en correspondencia con el algoritmo seleccionado para ello. En la clase Modelo_mineria_de_datos se brinda la flexibilidad de emplear el algoritmo de clasificación que se desee haciendo uso de la API de Weka. A continuación, se describe el funcionamiento de algunos de los principales métodos:

Class '**Interaccion_base_datos**':

Obtener_datos: función encargada de obtener datos de la Base de Datos Históricos del SCADA en cuestión para su posterior procesamiento. Realiza consultas del tipo *SELECT*.

Class '**Modelo_mineria_de_datos**':

Obtencion_de_datos: función que se apoya en la clase Interaccion_base_datos como interfaz para acceder a la información de la Base de Datos.

Crear_base_de_conocimientos: función donde se realiza el procesamiento de los datos y la creación de la base de conocimientos. Está función actúa como elemento nuclear, en donde se puede reimplementar la creación del algoritmo de clasificación que se desee.

Clasificar_tupla: constituye otra de las funciones más importantes, dado una cadena de texto en formato JSON se encarga de decodificar y clasificar según los parámetros, devolviendo un resultado consistente en la clasificación de esta tupla.

Conclusiones del capítulo

Una vez concluido el capítulo de la propuesta de solución se realizan las siguientes consideraciones:

Se analizó la pertinencia del uso del Módulo de Base de Datos Históricos de los sistemas SCADA para el proceso de extracción de conocimiento y descubrimiento de patrones arribando a conclusiones positivas.

Se realizó el proceso de Descubrimiento de Conocimiento en Bases de Datos con la ayuda de 4 subprocesos fundamentales: definición de objetivos del proceso, preparación de los datos, Minería de Datos y análisis. Dichos subprocesos ejecutaron 9 actividades en total.

Se utilizó a la plataforma Weka como herramienta de apoyo para el cumplimiento y ejecución de las tareas de Minería de Datos, y logrando una efectiva integración con el Sistema de Automatización Industrial basado en tecnologías libres de la Universidad de las Ciencias Informáticas.

CAPÍTULO 3 VALIDACIÓN DE LA PROPUESTA

En el presente capítulo se describe la validación de los resultados investigativos obtenidos y se explican además los métodos utilizados en este proceso. Para fundamentar la validación del modelo propuesto se realizaron pruebas funcionales a los módulos desarrollados luego de ser integrados al SAINUX, creando las bases de conocimiento con juegos de datos de distintos rangos de tiempo, variando desde una semana hasta el histórico correspondiente a un trimestre. Posteriormente fueron aplicadas otras técnicas como la escala de Likert, la cual se fundamenta en la opinión de expertos y especialistas del negocio para validar los modelos obtenidos y la propuesta de solución.

3.1 Validación de los modelos descriptivos desarrollados

Como se había mencionado anteriormente, al momento de desarrollar esta investigación, el Sistema de Automatización Industrial Basado en tecnologías libres desarrollado en la Universidad de las Ciencias Informáticas no se encuentra desplegado en un ambiente corporativo real, sino en ambientes de prueba, laboratorios de desarrollo y otros ambientes controlados. Por este motivo el proceso de pruebas y validaciones se ve dificultado parcialmente, al no contar con las características “hostiles” que surgen cotidianamente en un entorno de supervisión y control real. No obstante, se considera que con las pruebas que se pueden realizar bajo las condiciones del entorno actual es posible evaluar con la pertinencia adecuada la calidad del proceso de descubrimiento de datos llevado a cabo en esta investigación.

Para la validación de los modelos descriptivos se llevaron a cabo análisis sobre los resultados de los siguientes archivos como se muestra en la tabla 5:

Tabla 5: Archivos correspondientes a los datos del proceso de validación

Nombre de archivo	Tiempo	No de instancias
SAINUX_data_mining_model_5_10_016.arff	1 día	8690
SAINUX_data_mining_model_28_3_016_to_4_4_016.arff	1 semana	120956
SAINUX_data_mining_model_12_4_016_to_9_7_016.arff	3 meses	1555119

Como se puede observar el volumen de datos sobre el cual se realizan las validaciones llega a sobrepasar el millón de instancias en el último archivo presentado en la tabla 5, lo cual es un monto nada despreciable y a tono con la cantidad de eventos que suceden usualmente en los grandes centros de control y supervisión, en los cuales se pueden registrar hasta 40 000 eventos extraordinarios en el transcurso de un solo día. La concurrencia y velocidad de respuesta en cuanto a la creación de los modelos descriptivos también constituye un elemento a tener en cuenta, aun cuando no es el objetivo primordial de esta investigación.

Luego de ejecutar el algoritmo de reglas de asociación A-Priori para el primer archivo con una configuración de parámetros similar a la mostrada en el capítulo 2, se obtienen las siguientes reglas:

```
Best rules found:
1. day_seconds='(-inf-21599.75]' 2214 ==> description=alarma:alto-alto 1789    conf:(0.81)
 2. day_seconds='(-inf-21599.75]' resource_id='(-inf-7.25]' 2214 ==> description=alarma:alto-alto
1789    conf:(0.81)
 3. resource_id='(9.75-inf)' 4027 ==> description=alarma:alto 3246    conf:(0.81)
 4.  day_seconds='(21599.75-43196.5]'  resource_id='(9.75-inf)'  1360  ==>  description=alarma:alto
1096    conf:(0.81)
 5. day_seconds='(43196.5-64793.25]' 2117 ==> description=alarma:alto 1695    conf:(0.8)
 6.  day_seconds='(43196.5-64793.25]'  resource_id='(9.75-inf)'  2117  ==>  description=alarma:alto
1695    conf:(0.8)
 7. resource_id='(-inf-7.25]' 3087 ==> description=alarma:alto-alto 2376    conf:(0.77)
 8.  resource_id='(9.75-inf)'  variation_value='(1-495]'  1505  ==>  description=alarma:alto  1125
conf:(0.75)
 9.  resource_id='(9.75-inf)'  variation_value='(-493-1]'  1469  ==>  description=alarma:alto  1068
conf:(0.73)
10. day_seconds='(21599.75-43196.5]' 2233 ==> description=alarma:alto 1229    conf:(0.55)
```

Como se puede observar, el índice de confianza incluso en la regla 40 es apenas inferior al 50 por ciento (44 %). Se pueden apreciar de forma sencilla reglas que pueden resultar muy interesantes para el proceso de apoyo a la toma de decisiones. Por ejemplo, dada esta base de conocimientos, la regla número 1 muestra que cuando el momento del día se encuentra en el intervalo de 0 a 21599.75 (lo que es equivalente a el lapso de tiempo comprendido entre las 0 y 6 horas de la madrugada), los eventos críticos que ocurran serán alarmas de alta severidad (“alarma:alto-alto”) con un índice de confianza del 81 %. De igual forma se puede observar que en ese mismo lapso de tiempo un evento generado por los recursos con identificador comprendido entre 0 y 7 (-inf-7.25) generarán de igual forma alarmas de severidad alta con un índice de confianza del 81 %.

Como expresara (Orallo, 2006), la calidad de las reglas de asociación se ve en muchos casos ampliamente afectada con la presencia de atributos fuertemente descompensados. Como es el caso del atributo descripción, en las 20 mejores reglas obtenidas con el algoritmo A-Priori están relacionadas en su mayoría con alarmas de tipo “alto-alto” o “alto”, sin ilustrar casos de tipo “bajo” u otras variantes, precisamente dado porque el juego de datos con que fueron creadas las reglas de asociación no contemplan relaciones lo suficientemente descriptivas y con un índice de confianza lo suficientemente elevado para aparecer en las primeras 20 mejores reglas.

A continuación, se muestran los datos del resultado para la ejecución del algoritmo de agrupamiento *SimpleKMeans*, con una configuración de parámetros similar a la mostrada en el capítulo 2, incluyendo la configuración con 5 clústers:

Final cluster centroids:

		Cluster#				
Attribute	**Full Data**	**0**	**1**	**2**	**3**	**4**

	(8690.0)	(796.0)	(987.0)	(3490.0)	(1749.0)	(1668.0)
day_seconds	42826.5673	68597.5	53697.3799	25400.1338	51737.0492	51214.2566
resource_id	8.8611	8.4987	10.4792	6.49	10.8937	10.9065
previous_value	496.3692	484.4146	494.8602	499.4817	291.2047	711.5821
current_value	501.2522	495.1055	493.3161	499.6358	708.8856	294.5474
variation_value	4.8831	10.691	-1.5441	0.1542	417.681	-417.0348
description	alarma:alto	alarma:alto	alarma:bajo-bajo	alarma:alto-alto	alarma:alto	alarma:alto

En la tabla número 6 se muestran los resultados estadísticos de este agrupamiento:

Tabla 6: Resultado estadístico aplicación del algoritmo SimpleKMeans

No clúster	Cantidad de instancias	Porciento
0	796	9 %
1	987	11 %
2	3490	40 %
3	1749	20 %
4	0001668	19 %

En la siguiente figura se muestra la distribución de los grupos, permitiendo apreciar la excelente delimitación que se logra con la ejecución de este algoritmo de agrupamiento:

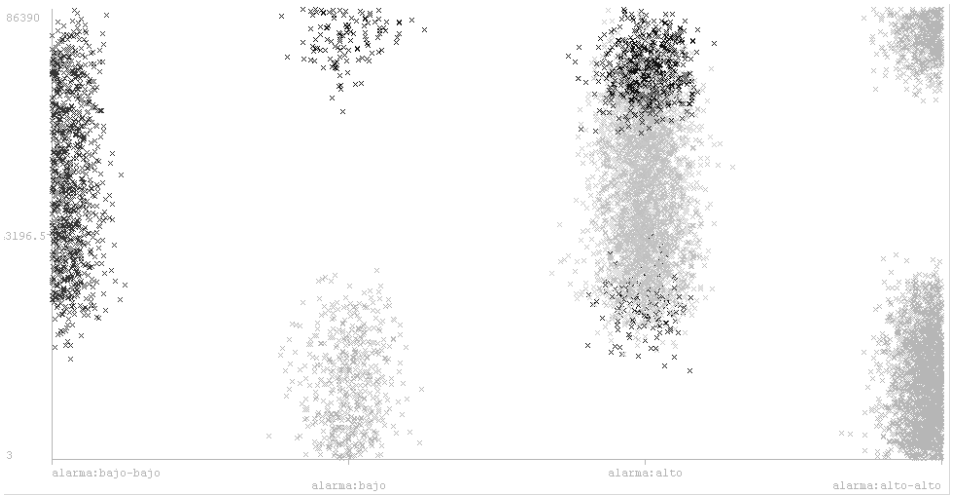


Figura 20: Visualización del agrupamiento luego de la aplicación del algoritmo SimpleKMeans

Para consultar el resultado del resto de las visualizaciones y tablas estadísticas de la creación de los modelos para los demás archivos seleccionados se debe consultar el **ANEXO 1**.

3.2 Validación del modelo predictivo desarrollado

Al realizar la ejecución del modelo de clasificación usando el algoritmo J48 de Weka se obtienen resultados positivos en cuanto al proceso de predicción de las instancias de cada archivo, trabajando en todos los casos con un porciento de certeza superior al 98 %, a continuación, se presentan los resultados de la ejecución de este modelo con el primer archivo, correspondiente a

un día de supervisión y control, usando la validación cruzada a 10 grupos. La matriz de confusión obtenida es la siguiente:

```
=== Confusion Matrix ===

   a    b    c    d   <-- classified as
 970    0    0    0 |    a = alarma:bajo-bajo
   1  527    3    0 |    b = alarma:bajo
  44   13 4753   52 |    c = alarma:alto
   0    0    0 2327 |    d = alarma:alto-alto
```

Figura 21: Matriz de confusión clasificación mediante el algoritmo por J48

Como se puede observar a simple vista, la diagonal principal adquiere la mayor cantidad de valores, significando que el índice de certeza es elevado, ya que esto significa que, en la matriz de confusión, la diagonal principal recoge a los elementos clasificados como “verdaderos positivos” o sencillamente clasificados de forma correcta. A continuación, se presenta el resumen estadístico de este modelo:

```
=== Summary ===

Correctly Classified Instances        8577               98.6997 %
Incorrectly Classified Instances       113                1.3003 %
Kappa statistic                          0.9785
Mean absolute error                      0.0119
Root mean squared error                  0.079
Relative absolute error                  3.9559 %
Root relative squared error             20.4085 %
Total Number of Instances             8690
```

Figura 22: Sumario del proceso de clasificación mediante el algoritmo J48

Se señala dentro del sumario el elevado por ciento de instancias clasificadas correctamente (superior a 98.6 %), como signo de que los datos con que cuenta el modelo brindan información importante y valiosa para el proceso de clasificación.

La siguiente imagen muestra una visualización de los errores cometidos en el proceso de clasificación, los errores son señalados con íconos en forma de cuadro a diferencia de los clasificados correctamente, los cuales se muestran en forma de punto:

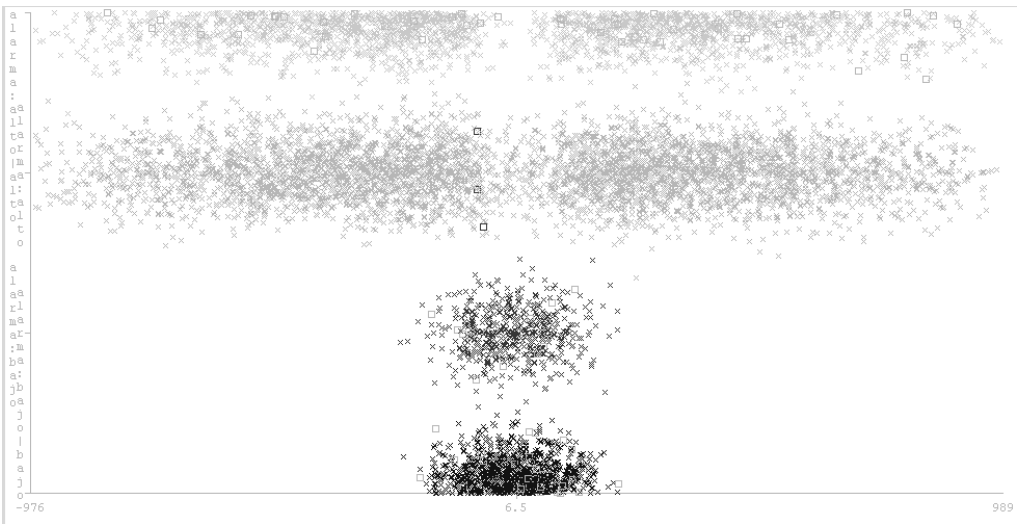


Figura 23: Visualización en el proceso de clasificación

Como se ha resaltado anteriormente, la evaluación de los modelos descriptivos no cuenta con un método determinante. Teniendo en cuenta lo planteado por (Hernández Orallo, 2008), un excelente criterio de evaluación es conocer si el modelo obtenido en la fase de aprendizaje presenta un comportamiento útil en el área de aplicación para el cual fue desarrollado. Dicho criterio presenta una estrecha relación con determinados factores como el interés que despierta, la novedad, su simplicidad, aplicabilidad y comprensibilidad. Por ello se hace necesaria la consulta de expertos para validar la importancia de los modelos obtenidos.

Con tal fin se decide realizar una encuesta a 11 especialistas del Centro de Informática Industrial de la UCI, con más de 5 años de experiencia en este trabajo, pero que su área de aplicación no se encuentre dentro de los softwares de supervisión y control de procesos industriales, de modo que no deben ser considerados especialistas funcionales del tema. A dichos especialistas se les presentan los resultados de esta investigación y luego se procede a llenar la encuesta (consultar **ANEXO 2**) con el objetivo de evaluar la validez y aplicabilidad de los modelos obtenidos. Posteriormente se calcula el índice porcentual o IP, capaz de integrar en un solo valor el coeficiente del criterio de los especialistas sobre los criterios propuestos. La composición de experiencia de los especialistas es la siguiente:

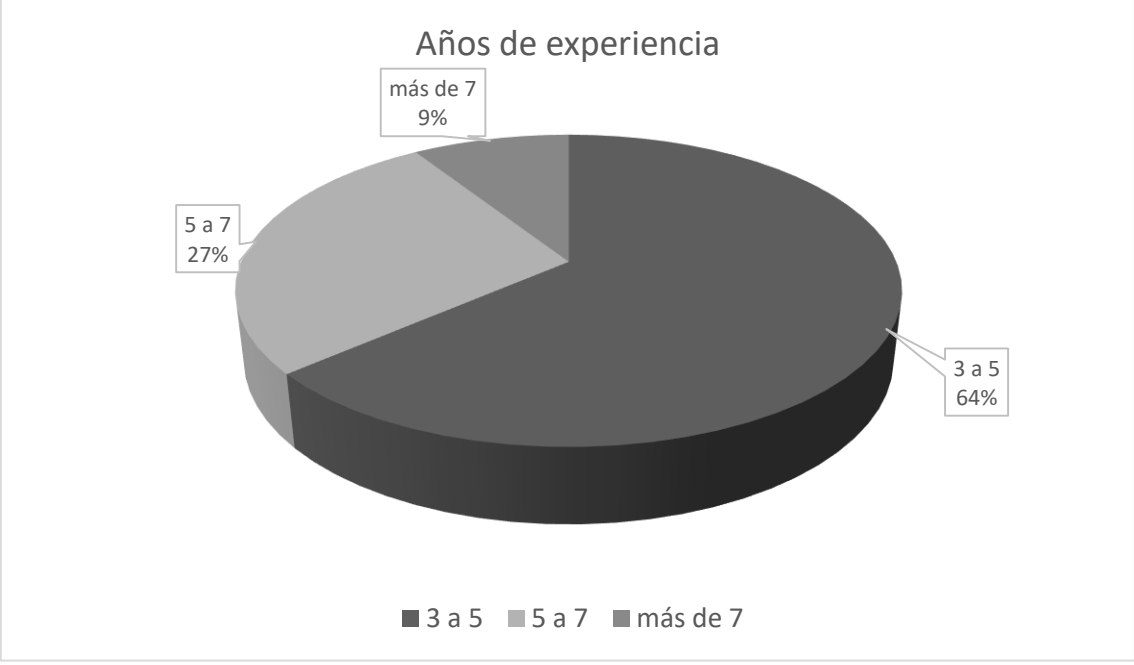


Figura 24: Experiencia de los especialistas a los que se realiza la encuesta.

El cálculo se realizó según la siguiente fórmula, adaptado a las categorías de evaluación previamente establecidas en la encuesta, donde 1 significa el valor mínimo y 5 la máxima calificación.

$$IP = \frac{5(\%\ MA) + 4(\%\ A) + 3(\%\ MnA) + 2(\%\ PA) + (\%\ I)}{5}$$

Donde:

MA: es Muy Adecuado

A: Adecuado

MnA: Medianamente Adecuado

PA: Poco Adecuado

I: Inadecuado

En la siguiente figura se muestra el IP correspondiente a cada uno de los criterios evaluados:

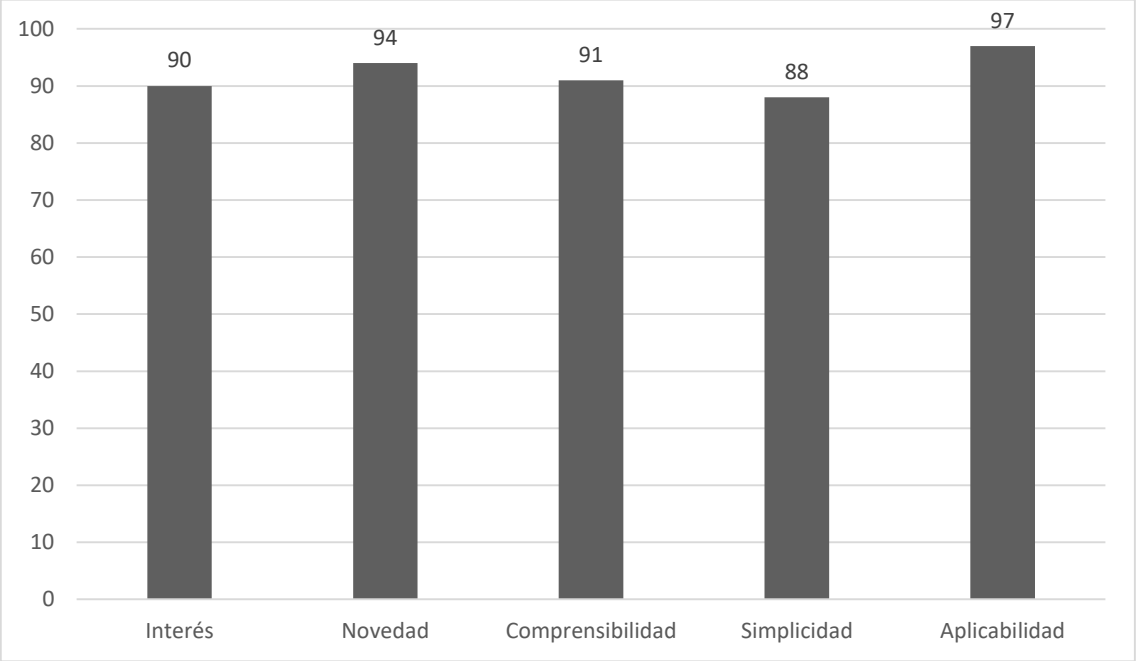


Figura 25: Resultados de la encuesta para validación de los modelos

3.3 Criterios de expertos y especialistas funcionales sobre la propuesta

El uso de las encuestas se ha esparcido considerablemente como herramienta de apoyo entre los especialistas e investigadores de varias ramas tanto de las ciencias técnicas como sociales. Partiendo de la premisa de que, para conocer y evaluar el comportamiento de determinados fenómenos, una de las vías más simples y quizás adecuadas consiste en aplicar encuestas a especialistas relacionados de alguna forma con el fenómeno en cuestión y que posean una experiencia considerable en el trabajo con este.

Uno de los aspectos fundamentales constituye la determinación del número mínimo de expertos a los cuales aplicar la encuesta, de modo que constituya un grupo lo suficientemente significativo de personas, de igual forma es muy importante la evaluación de estos especialistas como expertos. A juzgar por lo planteado por (García, 2008), la cantidad óptima de expertos a encuestar para la aplicación de una encuesta debe oscilar entre 15 y 25, criterio respaldado además por una

considerable cantidad de autores de la actividad docente investigativa y la experiencia de más de 30 años del autor del artículo antes mencionado.

La cantidad de expertos seleccionados fue de 16, los cuales presentan más de 3 años de experiencia en el trabajo, interacción y desarrollo con los sistemas de supervisión y control. A dicho grupo le fue detallados los pormenores de la presente investigación, mostrándole los resultados y futuras potencialidades. Posteriormente se les aplicó un cuestionario (consultar **ANEXO 3**), con el fin de evaluar la propuesta. De igual modo se le agregó una prueba de autoevaluación con el objetivo de evaluar las competencias de cada uno. Con dicha prueba se calcula el coeficiente de competencias (K), partiendo de contar con el coeficiente de conocimiento (Kc), y las fuentes de argumentación (Ka), empleando la siguiente fórmula:

$$K = \frac{Kc + Ka}{2}$$

Donde:

Kc: constituye el índice de conocimiento que posee el experto sobre el negocio, calculado sobre la valoración del propio experto en escala de 1 a 10 siendo 10 el valor máximo.

Ka: Coeficiente de argumentación de los criterios del experto, se obtiene a partir de la suma de los valores alcanzados a partir de los elementos de la tabla siguiente:

Tabla 7: Cálculo del grado de influencia de las fuentes de argumentación

Id	Fuentes de argumentación	Grados de influencia		
		Alto	Medio	Bajo
F1	Análisis teóricos	0.3	0.2	0.1
F2	Experiencia obtenida	0.5	0.4	0.3
F3	Trabajos de autores nacionales	0.05	0.04	0.03
F4	Trabajos de autores extranjeros	0.05	0.04	0.03
F5	Conocimiento propio del tema	0.05	0.04	0.03
F6	Intuición	0.05	0.04	0.03

Luego de realizado el cálculo de K, sus resultados se pueden evaluar de forma nominal según la siguiente tabla:

Tabla 8: Conversión del valor K a escala nominal

Coeficiente de competencia	Valor
Alto	0.8 < K < 1.0
Medio	0.5 < K < 0.8

Bajo	K < 0.5

En la siguiente tabla se muestra un resumen de los resultados de la evaluación de cada uno de los 16 expertos consultados. Quedando evidenciado que el grupo posee un coeficiente de competencia suficientemente alto, al tiempo que de los 16 especialistas encuestados 13 de ellos poseen un alto índice de competencias (81.25 %) y 3 un índice de competencias medio, respaldando la elección de expertos realizada, así como la brindando un mayor apoyo a sus opiniones.

Tabla 9: Cálculo de índice de competencias

Experto No	Autoeval.	Kc	F1	F2	F3	F4	F5	F6	Ka	K	Nivel
Exp 1	9	0.90	0.20	0.40	0.04	0.04	0.04	0.05	0.77	0.84	**Alto**
Exp 2	5	0.50	0.30	0.40	0.05	0.04	0.05	0.04	0.88	0.69	**Medio**
Exp 3	7	0.70	0.20	0.50	0.05	0.05	0.04	0.05	0.89	0.80	**Alto**
Exp 4	7	0.70	0.30	0.40	0.04	0.03	0.03	0.04	0.84	0.77	**Medio**
Exp 5	9	0.90	0.20	0.40	0.03	0.03	0.03	0.05	0.74	0.82	**Alto**
Exp 6	10	1.0	0.20	0.40	0.04	0.04	0.04	0.05	0.77	0.89	**Alto**
Exp 7	8	0.80	0.20	0.30	0.05	0.04	0.04	0.04	0.67	0.74	**Medio**
Exp 8	10	1.0	0.30	0.50	0.04	0.04	0.05	0.05	0.98	0.99	**Alto**
Exp 9	10	1.0	0.30	0.40	0.03	0.03	0.04	0.05	0.85	0.93	**Alto**
Exp 10	8	0.80	0.30	0.40	0.05	0.04	0.04	0.03	0.86	0.83	**Alto**
Exp 11	8	0.80	0.20	0.50	0.05	0.04	0.05	0.04	0.88	0.84	**Alto**
Exp 12	8	0.80	0.30	0.40	0.03	0.03	0.04	0.04	0.84	0.82	**Alto**
Exp 13	10	1.0	0.20	0.30	0.03	0.03	0.05	0.05	0.66	0.83	**Alto**
Exp 14	8	0.80	0.30	0.40	0.04	0.03	0.03	0.04	0.84	0.82	**Alto**
Exp 15	9	0.90	0.30	0.50	0.05	0.05	0.04	0.05	0.99	0.95	**Alto**
Exp 16	9	0.90	0.20	0.40	0.03	0.03	0.04	0.05	0.75	0.83	**Alto**

Al momento de aplicar este tipo de encuestas se tuvieron en cuenta determinadas normas para obtener resultados lo más confiables posibles y acordes con los criterios reales de los especialistas como su realización bajo el anonimato, de modo que los criterios de un especialista no se vieran afectados por los de otro, además de que cada experto defendió sus argumentos sin la posibilidad de que su opinión particular fuera conocida por otra persona que quien le realizó la entrevista.

Con el objetivo de ofrecer los resultados de las encuestas de la forma más ilustrativa posible se aplica la técnica conocida como escala de Likert, en la cual se otorga una puntuación entre 1 y 5 a cada objeto como se muestra en la siguiente tabla:

Tabla 10: Valores usando escala de Likert

Criterios			Puntuación
Muy adecuado	Muy importante	Sí	5
Adecuado	Importante	Para la mayoría de los casos	4
Medianamente adecuado	Medianamente importante	En algunos casos	3
Poco adecuado	Poco importante	En la minoría de los casos	2
Inadecuado	No importante	No	1

Posteriormente se procede a calcular el IP mostrado con anterioridad, integrando en un salo valor la aceptación del grupo de expertos acerca de la solución propuesta:

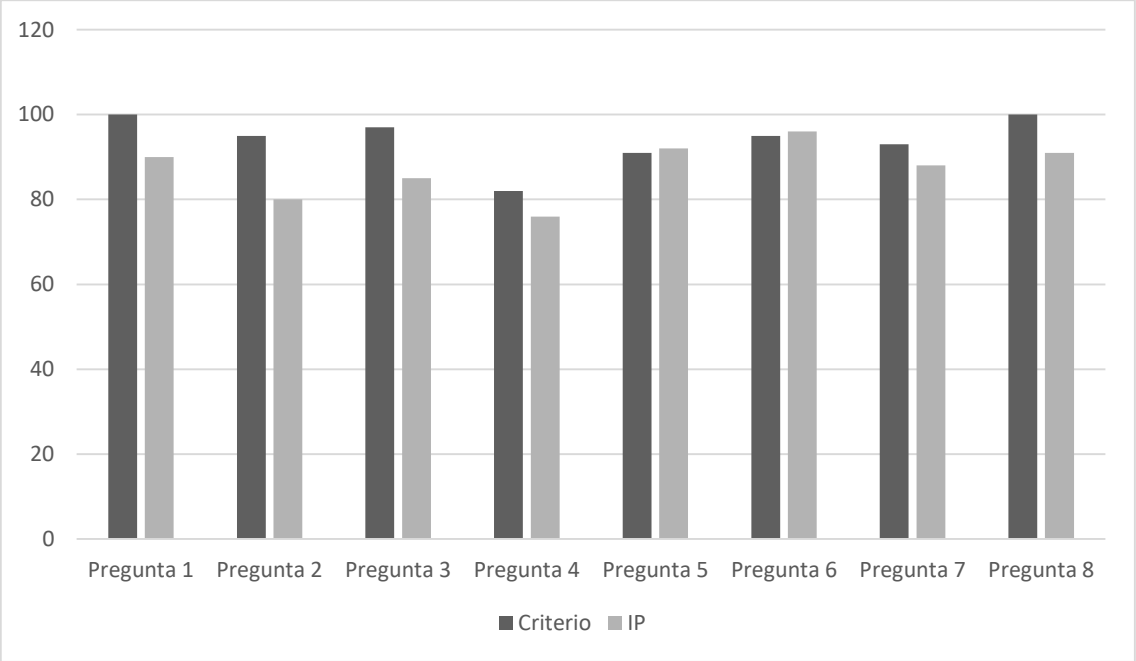


Figura 26: Datos obtenidos usando escala de Likert

Validación de los modelos predictivos desarrollados

3.4 Análisis del impacto económico y social de la solución propuesta

Con el surgimiento del Sistema de Automatización Industrial basado en tecnologías libres, la Universidad de las Ciencias Informáticas potencia el desarrollo de un software que permite encaminar esfuerzos hacia lograr la necesaria sustitución de importaciones en cuanto a los softwares de supervisión y control para ambientes industriales se refiere. Actualmente muchas empresas nacionales pagan costosas sumas por el contrato, soporte y actualización de sistemas de esta gama, los cuales es justo mencionar, ya poseen en la mayoría de los casos un marcado prestigio y apego a los estándares dictados por la Asociación de Equipamiento Ingenieriles y de Materiales de Usuario conocido como EEMUA, por lo que se hace necesario encaminar esfuerzos para que productos de factura nacional como el SAINUX aumenten sus capacidades funcionales

como: soportar una mayor cantidad de protocolos de comunicación con dispositivos de campo, flexibilidad ante el trabajo en ambientes industriales no controlados, certificación de los estándares que soportan los SCADA en la actualidad y entre otros se debe mencionar el aumento del nivel de gestión y apoyo a la toma de decisiones con módulos de inteligencia Artificial como el desarrollado en esta investigación.

Las técnicas de Minería de Datos aplicadas a cualquier software en sentido general implican un proceso muy costoso, partiendo de que requieren la recolección de los datos a minar, la preparación de la información, la correcta formulación del problema a resolver, la construcción de los modelos y finalmente la integración con el software, lo cual implica un conocimiento intelectual ya adquirido por parte de los especialistas en ciencia de datos. En términos económicos individuales se debe mencionar que el salario promedio de un especialista en MD ronda los 5 mil dólares mensuales como tasa mínima a juzgar por las encuestas publicadas por (KDnuggets, 2015). Al ser desarrollado este modelo ya no se hace necesaria la contratación de los servicios de un especialista de esta rama por las empresas nacionales o extranjeras que utilicen el SAINUX, si bien no es menos cierto que gradualmente se deben continuar enriqueciendo las funcionalidades de este modelo para lograr objetivos mucho más ambiciosos en cuanto al procesamiento inteligente, tarea que puede ser llevada a cabo partiendo de la base de esta investigación y manteniendo el conocimiento adquirido para su evolución.

Por otro lado, también es muy importante señalar que el desarrollo del SAINUX ha sido bajo los estándares del software libre, y al usar para el desarrollo de los módulos de los que consta el modelo herramientas libres como Weka (software bajo la licencia GPL), se ahorra la necesidad del pago de licencias y se continúa desarrollando un producto autóctono de Cuba, garantizando además la soberanía tecnológica del país.

A juzgar por (KDnuggets, 2015), el precio promedio de las licencias comerciales para softwares de descubrimiento de conocimiento y MD asciende a 16 000 dólares, algunos ejemplos conocidos son productos como:

- RapidMiner
- SAS base
- SAS Enterprise Miner
- IBM Anallytical Decision Management
- IBM SPSS Modeler Professional

Este elemento no constituye un problema, ya que la investigación desarrollada no implicaría gastos económicos para proseguir su desarrollo y comercialización, y en cambio dota al sistema de un elemento novedoso para el análisis de los procesos que se controlan.

Conclusiones del capítulo

Luego de ejecutar los métodos científicos para validar la propuesta de solución se concluye lo siguiente:

Al realizar las pruebas funcionales se validó el cumplimiento de los requerimientos objetivos planteados para el desarrollo de los módulos del modelo de alertas sobre la ocurrencia de eventos extraordinarios en SCADAS.

Tanto los especialistas funcionales como los desarrolladores de software que no tienen una estrecha vinculación con los sistemas de supervisión y control de procesos industriales mostraron una gran aceptación de la solución planteada evaluando indicadores como interés, novedad, aplicabilidad, comprensibilidad y simplicidad, con un índice por encima del 85 %.

Los expertos coincidieron en que el desarrollo de esta solución constituye un paso inicial de determinada importancia en cuanto al desarrollo futuro de sistemas inteligentes para el apoyo a la toma de decisiones a una escala mucho mayor.

Se analizó además la importancia económica que implica el desarrollo de este tipo de sistemas para el aumento de la competitividad en el mercado de softwares nacionales como el SAINUX. El ahorro en cuanto al pago de licencias y contratación de personal especializado en esta rama al usar herramientas libres para las tareas de MD.

Conclusiones generales

Al término del desarrollo de la presente investigación se puede arribar a las siguientes conclusiones:

Con el desarrollo de un modelo de alertas sobre la ocurrencia de eventos extraordinarios, para la prevención de eventos de alta criticidad se integran nuevas y modernas potencialidades al SAINUX, constituyendo un paso importante hacia el objetivo de aumentar el nivel de éste sistema respecto a los principales exponentes del estado del arte.

La solución propuesta incluye un modelo predictivo, con el cual se puede seguir trabajando para lograr un tratamiento mucho más eficiente en el proceso de control de alarmas ante la ocurrencia de eventos de alta concurrencia de eventos de carácter crítico en los centros de control y supervisión de ambientes industriales.

Mediante la realización de pruebas se pudo realizar la validación del cumplimiento de los objetivos de la solución informática implementada y apelando al criterio de la mayoría de los expertos, se arriba a la opinión de que la solución propuesta contribuye al control y supervisión de procesos que pueden resultar claves en los ambientes productivos empresariales.

La solución propuesta constituye un aporte social para nuestro país desde distintos puntos de vista, partiendo desde el eventual ahorro en cuanto a importaciones tanto de software, como licencias, contratación de personal extranjero especializado en temáticas como la ciencia de datos aplicada a la industria y demás.

El resultado obtenido puede ser usado como base para futuras investigaciones e implementaciones relacionadas con el área de control automático de procesos, extendiéndolo incluso a otras esferas de aplicación.

Recomendaciones

A partir de los resultados obtenidos en la presente investigación se recomienda:

Continuar desarrollando y enriqueciendo la solución planteada, partiendo de la base de la cual se dispone con esta, con el objetivo de llevar la solución y el SAINUX en general a planos mucho más competitivos en materia de sistemas de supervisión y control de procesos.

Evaluar por parte del grupo técnico del CEDIN, la incorporación de soluciones similares a los softwares desarrollados tanto en el centro como en la Universidad de las Ciencias Informáticas a modo general.

Continuar fomentando la investigación en los temas de ciencia de datos, descubrimiento de conocimiento e inteligencia artificial, aplicadas al control automático como base para futuros temas de maestrías y doctorados.

Incorporar soluciones de esta índole a sistemas sobre los cuales se investiga en el centro como la domótica, Internet de las Cosas y sistemas embebidos.

BIBLIOGRAFÍA

Abele, L., y otros. 2012. *Combining Knowledge Modeling and Machine Learning for Alarm Root Cause Analysis.* Department of Electrical Engineering and Information Technology,, Technische Universit¨at M¨unchen, Munich, Germany. Munich : s.n., 2012.

Acuna, E. 2008. *Unsupervised classification, Clustering.* Mayaguez, Puerto Rico : s.n., 2008.

Ahmadinejad, S.H. y Jalili, S. 2009. *Alert correlation using correlation probability estimation and time windows. Proc. of the Int. Conf. on Computer Technology and Development (ICCTD'09).* 2009.

Alfonso, V. y Keith, S. 2001. *Probabilistic alert correlation. Proc. of the 4th Int. Symposium on Recent Advances in Intrusion Detection.* 2001.

Álvarez Prados, V. 2009. *Bases de Datos Espacio - Temporales.pp 30-32.* 2009.

Amaya Torrado, Y. K., Barrientos Avendaño, E., & Heredia Vizcaíno, D. J. 2014. *Modelo predictivo de deserción estudiantil utilizando técnicas de minería de datos.* 2014.

ANSI/ISA-18.2. 2009. *Management of Alarm Systems for the Process Industries.* 2009. ISBN: 978-1-936007-19-6.

Antúnez, Y. 2015. Repositorio Institucional. [En línea] 21 de febrero de 2015. [Citado el: 20 de junio de 2016.] http://repositorio_institucional.uci.cu/jspui/handle/ident/8799.

Aragón Cáceres, J. A., Chávez Lorenzo, A., Pérez Javier, M., & Ravelo Hernández, L.A. 2011. *Servidor de Comunicación con sistemas externos del SCADA - UX.* s.l. : Serie Científica de la Universidad de las Ciencias Informáticas, 2011.

Bailey, D., & Wright, E. 2003. *Practical SCADA for Industry.* s.l. : Oxford: Newnes, 2003.

Berry, M. J., & Linoff, G. 2004. *Data mining techniques: for marketing, sales, and customer relationship management.* s.l. : John Wiley & Sons, 2004.

—. 1997. *Data mining techniques: for marketing, sales, and customer support.* s.l. : John Wiley & Sons, 1997.

Berzal, F. 2009. *Métodos de agrupamiento.* Granada, España : s.n., 2009.

Boyer, S. A. 2004. *SCADA: Supervisory Control and Data Adquisition.* s.l. : ISA - The Instrumentation, Systems, and Automation Society., 2004.

Bullemer, P., y otros. 2011. *Towards Improving Operator Alarm Flood Responses: Alternative Alarm Presentation Techniques.* Abnormal Situation Management Consortium. 2011.

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., & Zanasi, A. 1998. *Discovering data mining: from concept to implementation.* s.l. : Prentice-Hall, 1998.

Cabús, J., Navarrete, D. G. y Porras, R. P. 2004. *Sistemas SCADA.* Catalunya : s.n., 2004.

Calderón Méndez, N. 2010. *MINERÍA DE DATOS.UNA HERRAMIENTA PARA LA TOMA DE DECISIONES.* Guatemala : s.n., 2010.

Cerón Reyes, M. y Gómez Díaz, H. 2010. *Bases de Datos.Minería de.* 2010.

DELGADO, A. 2006. *Desarrollo de aplicaciones con enfoque SOA.* 2006.

EEMUA 191. 2007. *Alarm Systems, A Guide to Design, Management and Procurement.* 2007. ISBN: 0-85931-155-4.

Fayyad, U., Haussler, D., & Stolorz, P. E. 1996. *KDD for Science Data Analysis: Issues and Examples.* 1996.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth. 1996a. *From data mining to knowledge.* s.l. : AI Magazine, 1996a.

Gallardo Arancibia, J. 2008. *Metodología para el Desarrollo de Proyectos en Minería de Datos CRISP-DM.* 2008.

García González, F. J. 2013. *Minería de Datos a datos obtenidos por el Centro Andaluz de Medio Ambiente (CEAMA).* Granada : s.n., 2013.

García, L., & Fernández, S. J. 2008. Procedimiento de aplicación del trabajo creativo en grupo de expertos. 2008, págs. 46-50.

Garre, M., Cuadrado, J. J., Sicilia, M. A., Rodríguez, D., & Rejas, R. 2007. *Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software.* Madrid : Revista Española de Innovación, Calidad e Ingeniería de Software, 2007. págs. 6-22.

General Electric Company. 2015. Proficy HMI/SCADA - iFIX. [En línea] 2015. http://www.ge-ip.com/products/proficy-hmi-scada-ifix/p3311.

Hand, D. J., Mannila, H., & Smyth, P. 2001. *Principles of data mining.* s.l. : MIT press, 2001.

Haque, M. Tarafdar y Kashtiban, A.M. 2007. *Application of Neural Networks in Power Systems; A Review.* 2007.

Hasperué, W. 2012. *Extracción de conocimiento en grandes bases de datos utilizando estrategias adaptativas.* Buenos Aires, Argentina : Universidad Nacional de la Plata, Facultad de informática, 2012.

Hernández Orallo, J. 2008. *Técnicas de Minería de Datos.* Valencia : s.n., 2008.

Hu, H.-W., Chen, Y.-L., & Tang, K. 2009. *A dynamic discretization approach for constructing decision trees with a continuous label.* s.l. : Knowledge and Data Engineering, IEEE Trnasactions, 2009.

Hua, H., & Zhao, H. 2009. *A Discretization Algorithm of Continuous Attributes Based on Supervised Clustering.* s.l. : CCPR 2009. Chinese Conference, 2009.

Hyper scada server. 2014. Hyper scada server. [En línea] 2014. [Citado el: 19 de 1 de 2015.] http://scadaserver.com/software/ht3-scada-software-reports.php.

Ian, H. y Eibe, F. 2005. *Data Mining: Practical Machine Learning Tools and Techniques.* 2005.

Idaho National Laboratory. 2010. *Alarm System Research Plan.* US. Department of Energy. Idaho : s.n., 2010.

Jakobson, G. y Weissman, M. 1995. *Real-time telecommunication network management extending event correlation with temporal constraints.* 1995.

Karnouskos, S., & Colombo, A. 2011. Architecting the next generation of service-based SCADA/DCS system of systems-IECON 2011 - 37th Annual Conference. [En línea] 2011.

KDnuggets. 2015. *Annual Income/Salary for Analytics, Data Mining, Data Science Professionals Poll.* 2015.

Lara, A. T. 2011. *Marco de descubrimiento de conocimiento para datos estructuralmente complejos con énfasis en el análisis de eventos en series temporales.* Madrid, España : s.n., 2011.

Lee, K., y otros. 2008. *DDoS attack detection method using cluster analysis.* 2008.

Machine Learning Wiki. 2016. ML Wiki. [En línea] 12 de 10 de 2016. http://mlwiki.org/index.php/CRISP-DM.

Martín, C. E. R., & Arias, Y. E. P. 2015. *Identificación de sismos similares haciendo uso de técnicas de Minería de Datos.* La Habana, Cuba : s.n., 2015.

Martínez, R. 2014. http://www.postgresql.org. [En línea] PosgreSQL, 2 de 5 de 2014. [Citado el: 2 de 5 de 2014.] http://www.postgresql.org.es/sobre_postgresql.

Medina, G. Cuenca y Ortiz, C. 2010. *La IA y los retos de la minería de Datos.* Ecuador : s.n., 2010.

Microsoft. 2016. Conceptos de minería de datos. *msdn.microsoft.com.* [En línea] 2016. [Citado el: 7 de 7 de 2016.] https://msdn.microsoft.com/es-es/library/ms174949.aspx.

Moine, J. M., Haedo, A. S. y Gordillo, S. 2010. *Estudio comparativo de metodologías para minería de datos.* Buenos Aires. : s.n., 2010.

Molina López, J. M. and García Herrero , J. 2006. *TÉCNICAS DE ANÁLISIS DE DATOS.APLICACIONES PRÁCTICAS UTILIZANDO MICROSOFT EXCEL Y WEKA.* 2006.

Montero, D., Barrantes, D. B., & Quiros, J. M. 2004. *Introducción a los sistemas de Supervisión, Control y Adquisisción de Datos.* Costa Rica : s.n., 2004.

Montes de Oca, O. 2015. *SOLUCIÓN INFORMÁTICA PARA EL DESCUBRIMIENTO DE CONOCIMIENTO EN LOS REGISTROS DE LA PLATAFORMA EDUCATIVA NAVIGO.* La Habana, Cuba : Universidad de las Ciencias Informáticas, 2015.

Navarro Marset, R. 2015. *RESST vs Web Services.* 2015.

O. Aizpurúa, R. Galán, A. Jiménez. 2010. *Análisis de alarmas masivas en los centros de despacho de energía eléctrica utilizando técnicas de inteligencia artificial y ontologías.* 2010.

Obregón Neira, N. 2008. *Un Modelo Conceptual para el Desarrollo de Árboles de.* 2008.

Orallo, J. H., Quintana, M. J. R., & Ramírez, C. F. 2006. *Práctica de Minería de Datos Introducción al Weka.* Valencia, España : s.n., 2006.

Pemex. 2012. Manual de Operación de la Franquicia Pemex. [En línea] 2012. [Citado el: 2 de 1 de 2015.] http://www.ref.pemex.com/files/content/02franquicia/sagli002/sagli002_10d.html.

Pérez Nieblas, L. y Manresa Sánchez, A. 2014. *Sistema de inteligencia de negocios para la Oficina Nacional de Estadísticas e Información.* La Habana : s.n., 2014.

Pérez, I., & León, B. 2007. *Lógica difusa para principiantes. .* Caracas, Venezuela : UCAB, 2007.

Piatetsky, G., & Frawley, W. 1991. *Knowledge discovery in databases.* s.l. : MIT press, 1991.

Qin, X y Lee, W. 2003. *Statistical causality analysis of infosec alert data. Proc. of the 6th Int. Symposium on Recent Advances in Intrusion Detection.* 2003.

Riquelme, J. C., Ruiz, R. and Rodríguez, D. 2009. *Finding Defective Software Modules by Means.* Sevilla : s.n., 2009.

Rodríguez Penin, A. 2007. *Sistemas SCADA – Guía Práctica.* España : Marcombo, 2007.

Rodríguez Penin, A. 2012. *Sistemas SCADA. Tercera Edición. .* España : Marcombo, 2012.

Rodríguez Suárez, Y. y Díaz Amador, A. 2003. *Herramientas de Minería de Datos.* La Habana : s.n., 2003.

Rodríguez, E. P. 2014. *Descubrimiento de conocimiento a partir de la relación rasgos de la personalidad-rendimiento laboral en proyectos informáticos.* La Habana, Cuba : Universidad de las Ciencias Informáticas, 2014.

Romeu Gullart, P. y Pardo Albiach, J. 2010. *MINERÍA DE DATOS APLICADA AL ANÁLISIS DEL TRATAMIENTO INFORMATIVO DE LA DROGADICCIÓN.* 2010.

Ruiz Vera, Z. y Plasencia Salgueiro, A. 2013. *AGRUPAMIENTO JERÁRQUICO Y DE K -MEANS.* La Habana : s.n., 2013.

Salah, S, Maciá-Fernández, G. y Díaz-Verdejo, J. 2013. *A model-based survey of alert correlation techniques.* s.l. : Computer Networks, ELSEVIER, 2013.

Sánchez Corales, Y. y Dávila Hernández, F. 2012. *Técnicas de Minería de datos aplicadas al diagnóstico de entidades clínicas.* La Habana : s.n., 2012.

Siemens Industry, Inc. 2010. *Setting a new standard in alarm management.* Siemens. 2010.

Tanna, P., & Ghodasara, Y. 2014. *Using Apriori with WEKA for Frequent Pattern Mining.* s.l. : IJETT, 2014.

Universidad de Oviedo. 2007. Área de Ingeniería de Sistemas y Automática. [En línea] Universidad de Oviedo, 10 de 12 de 2007. http://isa.uniovi.es/~felipe/files/infindII/documentos/scadas.pdf.

Vallejos, S. J. 2006. *Minería de Datos.* Buenos Aires. Argentina : s.n., 2006.

Vega Torres, L. y Rojas Díaz, L. 2008. *La Inteligencia de Negocio.Su implemntación mediante la plataforma Pentaho.* La Habana : s.n., 2008.

Villena Román, J., Crespo García, R. y García Rueda, J. J. 2011. *Minería de Datos.Inteligencia en Redes de Comunicaciones. p 4-7.* Madrid : s.n., 2011.

Wilford Rivera, I., Rosete Suárez, A. y Rodríguez Díaz, A. 2009. *Análisis de Información Clínica mediante técnicas de Minería de Datos.* La Habana : s.n., 2009.

Winer, D. 2001. *Heads up: A key difference between SOAP and XML-RPC.* 2001.

Wonderware. 2015. *Datasheet Historian.* 2015.

Zabre, E. y Gómez, O. 2012. *Panorama de la racionalización de sistemas de alarmas en el sector industrial y eléctrico.* México : s.n., 2012.

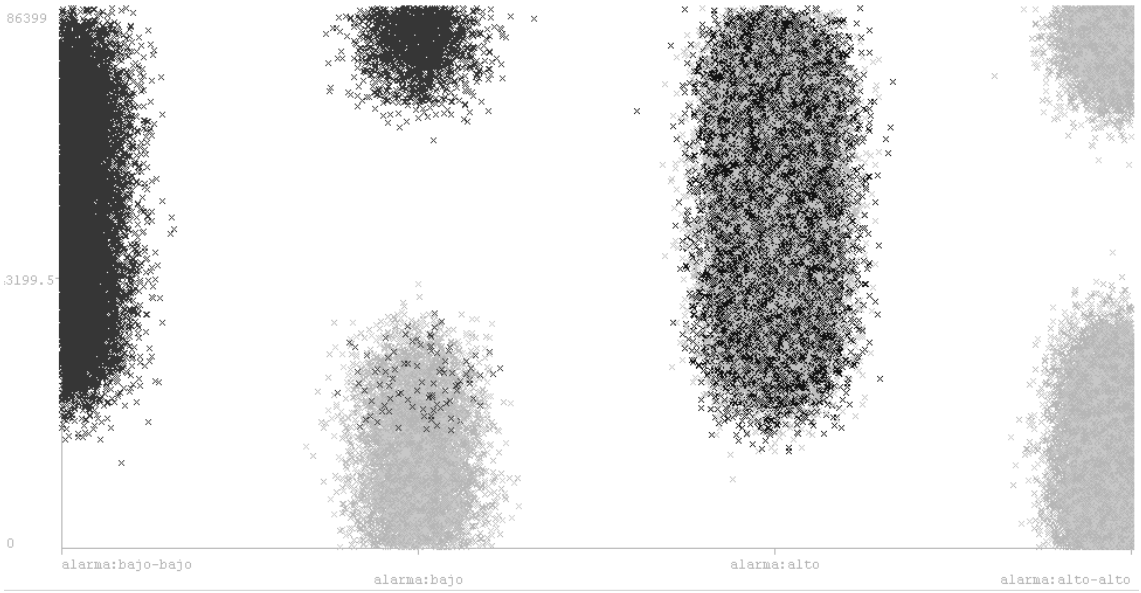
ANEXO 1: EJECUCIÓN DE ALGORITMO A-Priori SOBRE EL ARCHIVO SAINUX_DATA_MINING_MODEL_28_3_016_TO_4_4_016.ARFF, DE 120964 INSTANCIAS

```
Best rules found:

 1. day_seconds='(43199.5-64799.25]' 30211 ==> description=alarma:alto 24506    <conf:(0.81)>
lift:(1.72) lev:(0.08) [10229] conv:(2.79)
 2. day_seconds='(-inf-21599.75]' 30093 ==> description=alarma:alto-alto 24179    <conf:(0.8)>
lift:(2.4) lev:(0.12) [14102] conv:(3.38)
 3. description=alarma:alto-alto 40503 ==> day_seconds='(-inf-21599.75]' 24179    <conf:(0.6)>
lift:(2.4) lev:(0.12) [14102] conv:(1.86)
 4. resource_id='(8.25-inf)' 29503 ==> day_seconds='(21599.75-43199.5]' 16320    <conf:(0.55)>
lift:(2.22) lev:(0.07) [8979] conv:(1.68)
 5. day_seconds='(21599.75-43199.5]' 30095 ==> resource_id='(8.25-inf)' 16320    <conf:(0.54)>
lift:(2.22) lev:(0.07) [8979] conv:(1.65)
 6. day_seconds='(64799.25-inf)' 30557 ==> description=alarma:alto 16494    <conf:(0.54)>
lift:(1.14) lev:(0.02) [2054] conv:(1.15)
 7. day_seconds='(21599.75-43199.5]' 30095 ==> description=alarma:alto 16158    <conf:(0.54)>
lift:(1.14) lev:(0.02) [1936] conv:(1.14)
 8. resource_id='(-inf-2.75]' 27240 ==> description=alarma:alto 14360    <conf:(0.53)>
lift:(1.12) lev:(0.01) [1487] conv:(1.12)
 9. resource_id='(2.75-5.5]' 30853 ==> description=alarma:alto 16161    <conf:(0.52)> lift:(1.11)
lev:(0.01) [1581] conv:(1.11)
10. resource_id='(5.5-8.25]' 33360 ==> description=alarma:alto 16856    <conf:(0.51)> lift:(1.07)
lev:(0.01) [1091] conv:(1.07)
```

Aplicación del algoritmo de agrupamiento ***SimpleKMeans*** al archivo anterior:

Visualización del agrupamiento:



Resultado estadístico:

No clúster	Cantidad de instancias	Porciento
0	28599	24 %
1	15589	13 %
2	24408	20 %
3	23801	20 %
4	28559	24 %

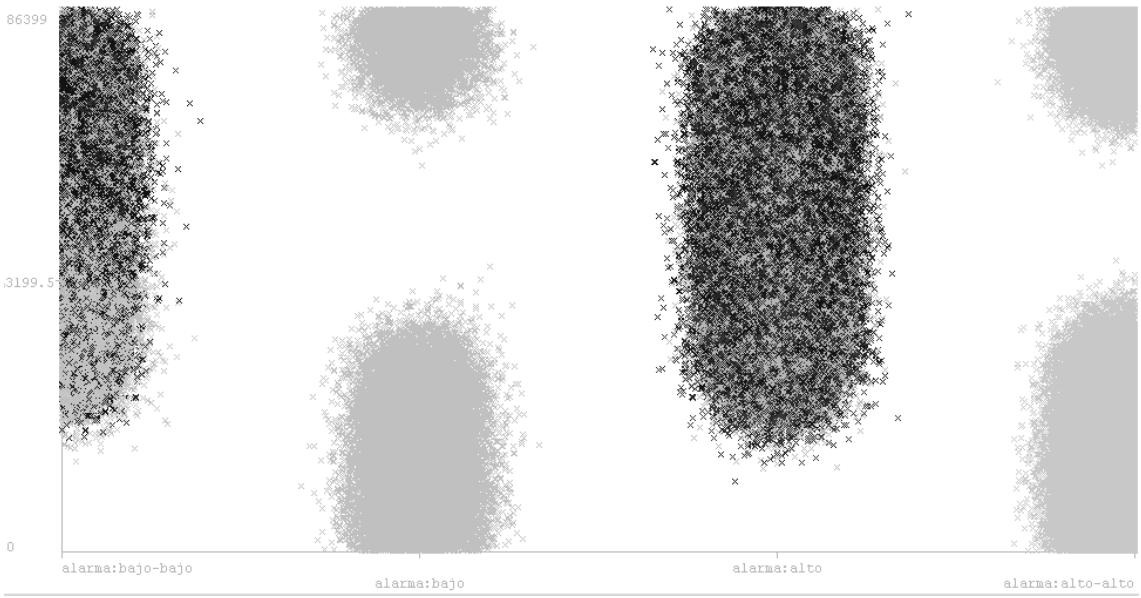
Ejecución de algoritmo A-Priori sobre el archivo *SAINUX_data_mining_model_12_4_016_to_9_7_016.arff*, de 1555119 instancias.

```
Best rules found:

1. day_seconds='(43199.5-64799.25]' 388952 ==> description=alarma:alto 314753    <conf:(0.81)>
lift:(1.71) lev:(0.08) [131198] conv:(2.77)
 2. day_seconds='(-inf-21599.75]' 388540 ==> description=alarma:alto-alto 313594    <conf:(0.81)>
lift:(2.4) lev:(0.12) [182836] conv:(3.44)
 3. description=alarma:alto-alto 523352 ==> day_seconds='(-inf-21599.75]' 313594    <conf:(0.6)>
lift:(2.4) lev:(0.12) [182836] conv:(1.87)
 4. day_seconds='(64799.25-inf)' 389059 ==> description=alarma:alto 209889    <conf:(0.54)>
lift:(1.14) lev:(0.02) [26283] conv:(1.15)
 5. day_seconds='(21599.75-43199.5]' 388568 ==> description=alarma:alto 209252    <conf:(0.54)>
lift:(1.14) lev:(0.02) [25878] conv:(1.14)
 6. resource_id='(5.5-8.25]' 387695 ==> description=alarma:alto 202219    <conf:(0.52)>
lift:(1.11) lev:(0.01) [19257] conv:(1.1)
 7. previous_value='(749.25-inf)' 388717 ==> variation_value='(-499.5-0]' 194691    <conf:(0.5)>
lift:(1.33) lev:(0.03) [48729] conv:(1.25)
 8. current_value='(249.75-499.5]' 389264 ==> variation_value='(-499.5-0]' 194936    <conf:(0.5)>
lift:(1.33) lev:(0.03) [48768] conv:(1.25)
 9. current_value='(-inf-249.75]' 387636 ==> variation_value='(-499.5-0]' 193844    <conf:(0.5)>
lift:(1.33) lev:(0.03) [48288] conv:(1.25)
10. previous_value='(499.5-749.25]' 387812 ==> variation_value='(-499.5-0]' 193895
<conf:(0.5)> lift:(1.33) lev:(0.03) [48273] conv:(1.25)
```

Aplicación del algoritmo de agrupamiento ***SimpleKMeans*** al archivo anterior:

Visualización del agrupamiento:



Resultado estadístico:

No clúster	Cantidad de instancias	Porciento
0	188361	12 %
1	347794	22 %
2	620645	40 %
3	201332	13 %
4	196987	13 %