

Universidad de las Ciencias Informáticas



Facultad 5

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

Automatización del análisis de Bases de Datos de Configuración de SCADA para  
poblar un Almacén de Datos.

Autor: Deyanira Bonaparte Pérez  
Tutor(es): Ing. Dayrien Corrales Díaz

La Habana, 2015  
"Año 57 de la Revolución"

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del presente trabajo y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Firma del Autor.  
Deyanira Bonaparte Perez

\_\_\_\_\_  
Firma del Tutor.  
Ing. Dayrien Corrales Diaz

## **DATOS DE CONTACTO**

**Tutor: Ing.** Dayrien Corrales Díaz

Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas, Cuba. Especialista General en Ciencias Informáticas con seis años de graduado y tres años de experiencia en el tema. Realizando especialidad de analista de datos en la maestría de Cibernética Aplicada, mención en Minería de Datos, Instituto de Cibernética, Matemática y Física (ICIMAF).

**Correo Electrónico:** dcorrales@uci.cu

## AGRADECIMIENTOS

“El hombre nunca sabe de lo que es capaz hasta que lo intenta”. Luego de tantos años de esfuerzo y estudio, hoy me gradué de Ingeniera en Ciencias Informáticas. A lo largo de este arduo recorrido, hay muchas personas que me han brindado su apoyo y ayuda, inspirándome a seguir hacia adelante. Hoy la vida me da la oportunidad de hacerles saber cuan agradecida estoy.

Mis primeros agradecimientos van para mis padres por quererme tal y como soy, por confiar en mí y brindarme todo su amor y cariño. A ustedes gracias por siempre estar a mi lado.

A mi mamá por ser mi inspiración, por demostrarme que una mujer puede ser trabajadora, inteligente, entregada y buena amiga, sin dejar de ser la mejor madre del mundo.

A mi papá por darme la oportunidad de tener un padre responsable, que me quiere y se siente orgulloso de su hija.

A mi hermano por quererme tanto como yo lo quiero a él y brindarme su apoyo cuando lo necesito. Por siempre tener una sonrisa para mí, es decir por siempre enseñarme su lado bueno de la luna.

A mi tío Jose Raúl y a mi tía Lisbeth por mantener su cariño y apoyo hacia mí, por estar presentes y formar parte de mi vida.

A mi primita Dayana por ser mi compañera de las causas imposibles, por ser mi amiga y confidente y por tantos años de fiesta.

Le agradezco en general a toda mi familia porque de una manera u otra me han inculcado educación y principios éticos, cualidades que han hecho de mí lo que hoy soy, una ingeniera en ciencias informáticas.

Quiero agradecerle de manera muy especial a Tito, mi tutor por guiarme, ayudarme y sobre todo por su paciencia y dedicación. Dayrien gracias por convertirte también en un buen amigo.

A mis dos grandes amigas Adriana y Tahimí, por su amistad incondicional durante tantos años, hoy no están presentes por cuestiones personales pero ustedes saben que las quiero mucho.

A Grethel, Julie y Leyanis por demostrarme ser muy buenas amigas y por seguir mi ejemplo y no rendirse, espero dentro de muy poco tiempo poder estar presente en sus tesis.

A todas las personas que he conocido a lo largo de estos cinco años en la universidad y han compartido conmigo desde el estudio hasta las borracheras del Mariposa: Rosmery, Esmerida, Maiyara, Javier, Chao, el Chino en fin a los 300.

A todos los profesores que han contribuido en mi formación como ingeniera y a los miembros de mi grupo 5402.

A todos los que se preocupan por mí, gracias.

## **DEDICATORIA**

Les dedico mi tesis a mi mamá y a mi papá por ser el motor impulsor de mis buenas decisiones y por darme las fuerzas suficientes para llegar hasta aquí. Pero no quiero terminar sin antes dedicarles este logro a mis abuelas Gladys y Deyanira y a mi abuelo Orlando, a ustedes muchas gracias por cuidarme desde el cielo.

## RESUMEN

A partir de la motivación de utilizar los datos generados en el proceso de producción controlados por los sistemas de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés), desarrollados en el Centro de Informática Industrial (CEDIN); se creó un Almacén de Datos para fomentar un ambiente de Inteligencia de Negocio, con el fin de generar análisis para la toma de decisiones. El resultado de estos análisis tiene una dependencia directa de las dimensiones que se establecen en el almacén.

La obtención de la fuente de datos para poblar las dimensiones, en algunos casos, requiere el empleo de herramientas y técnicas que permitan la unión de la característica de los datos existentes y el conocimiento empírico de los usuarios, para la obtención de una fuente de datos alternativa en el proceso de Extracción, Transformación y Carga (ETL, por sus siglas en inglés).

El uso de técnicas basadas en Minería de Datos, aún no es suficiente debido al tiempo que se necesita para estas tareas, y no se garantiza una integración directa al proceso de carga del almacén. En el presente trabajo se tiene como objetivo principal la automatización del análisis de la Base de Datos de Configuración del SCADA, para generar la fuente de datos que se requiere en el proceso de ETL hacia la dimensión Equipo Industrial del Almacén de Datos. Para ello se logra la integración del algoritmo de agrupamiento Simple K-means, mediante el uso de WEKA, en una herramienta que minimiza el tiempo requerido para este tipo de tareas.

Palabras claves: Agrupamiento, Almacén de Datos, ETL, Inteligencia de Negocio, SCADA, WEKA.

## Tabla de contenido

INTRODUCCIÓN.....	11
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....	16
Introducción .....	16
1.1 Inteligencia de Negocio .....	16
1.2 Proceso de extracción de conocimiento.....	18
1.2.1 Fases del KDD .....	19
1.2.2 Las metas del KDD .....	19
1.3 Minería de Datos. Características y objetivos .....	20
1.3.1 Técnicas de Minería de Datos .....	20
1.3.2 Tareas de Minería de Datos .....	22
1.4 Algoritmos de Minería de Datos .....	23
1.5 Base de Datos, Almacenes de Datos y Vista Minable .....	25
1.6 Pentaho Data Integration.....	27
1.7 WEKA .....	29
1.8 Conclusiones del Capítulo .....	30
CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN .....	31
2.1 Propuesta de solución .....	31
2.2 Modelo del Dominio .....	32
2.3 Funcionalidades del sistema .....	33
2.4 Historias de Usuario .....	35
2.5 Selección de la tecnología.....	41
2.5.1 Modelo Cliente-Servidor.....	41
2.5.2 WebSockets .....	42
2.5.3 JSON.....	43
2.5.4 Algoritmo de Minería de Datos seleccionado .....	44
2.5.5 Herramienta seleccionada para la Minería de Datos.....	46
2.5.6 Lenguaje de programación .....	46
2.5.7 Entorno de desarrollo.....	48
2.5.8 Metodología de Desarrollo de Software.....	49
2.5.9 Herramienta CASE.....	50
2.5.10 Patrones de diseño .....	50
2.7 Diagrama de Clases .....	51
2.8 Arquitectura de la solución .....	52
2.9 Diagrama de Colaboración .....	53
2.10 Conclusiones del Capítulo .....	54
CAPÍTULO 3: CONSTRUCCIÓN Y VALIDACIÓN.....	55
3.1 Diagrama de Componentes.....	55
3.1.2 Diagrama de Despliegue .....	56

3.2 Estándar de código .....	56
3.3 Validación de funcionalidades .....	58
3.4 Pruebas y resultados obtenidos .....	60
3.4.1 Plan de pruebas y Diseños de casos de pruebas .....	60
3.5 Pruebas de Aceptación .....	70
3.6 Conclusiones del Capítulo .....	74
Conclusiones Generales .....	75
RECOMENDACIONES .....	76
REFERENCIAS BIBLIOGRÁFICAS.....	44
BIBLIOGRAFÍA.....	46
Glosario de Términos .....	46
Anexos.....	48

## ÍNDICE DE FIGURAS

Figura 1. Fases del proceso de extracción de conocimiento en bases de datos.....	18
Figura 2. Representación gráfica de regresión lineal.....	23
Figura 3. Algoritmo K-means.....	23
Figura 4. Selector de interfaces de la herramienta WEKA.....	29
Figura 5. Diagrama de Modelo del Dominio.....	31
Figura 6. Modelo cliente-servidor.....	42
Figura 7. Ejemplo de Comunicación mediante Websocket.....	43
Figura 8. Diagrama de un sistema de coordenadas cartesianas.....	45
Figura 9. Diagrama de Clases.....	51
Figura 10. Arquitectura 4 capaz.....	52
Figura 11. Diagrama de Colaboración.....	53
Figura 12. Diagrama de Componentes.....	54
Figura 13. Diagrama de Despliegue .....	55
Figura 14. Archivo con formato arff para el uso de la herramienta WEKA.....	58
Figura 15. Resultado del Algoritmo K-means en la herramienta WEKA utilizando la distancia Euclidiana.....	58
Figura 16. Visualización de la asignación del clúster utilizando la distancia Euclidiana.....	59



## ÍNDICE DE TABLAS

Tabla 1: Requisitos funcionales del Servidor de análisis.....	34
Tabla 2: Historia de usuario número 1. Adicionar sesión.....	35
Tabla 3: Historia de usuario número 2. Eliminar sesión. ....	54
Tabla 4: Historia de usuario número 3. Insertar tareas.....	36
Tabla 5: Historia de usuario número 5. Eliminar tareas.....	37
Tabla 6: Historia de usuario número 6. Crear notificaciones. ....	37
Tabla 7: Historia de usuario número 8. Eliminar notificaciones. ....	38
Tabla 8: Historia de usuario número 9. Crear diccionario.....	38
Tabla 9: Historia de usuario número 10. Ampliar el ámbito de análisis. ....	39
Tabla 10: Historia de usuario número 11. Ejecutar Algoritmo de análisis. ....	39
Tabla 11: Historia de usuario número 12. Etiquetar grupos. ....	40
Tabla 12: Historia de usuario número 13. Enviar resultados. ....	41
Tabla 13: Plan de prueba para la Historia de Usuario 1. Adicionar sesión. ....	61
Tabla 14: Plan de prueba para la Historia de Usuario 2. Eliminar sesión. ....	62
Tabla 15: Diseño de caso de prueba para la Historia de Usuario 2. Eliminar sesión.....	62
Tabla 16: Plan de prueba para la Historia de Usuario 3. Insertar Tarea. ....	63
Tabla 17: Diseño de caso de prueba para la Historia de Usuario 3. Insertar Tarea.....	64
Tabla 18: Plan de prueba para la Historia de Usuario 5. Eliminar Tarea.....	64
Tabla 19: Diseño de caso de prueba para la Historia de Usuario 5. Eliminar Tarea.....	65
Tabla 20: Plan de prueba para la Historia de Usuario 6. Crear Notificaciones. ....	65
Tabla 21: Diseño de caso de prueba para la Historia de Usuario 6. Crear Notificaciones.....	66
Tabla 22: Plan de prueba para la Historia de Usuario 8. Eliminar Notificaciones. ....	66
Tabla 23: Diseño de caso de prueba para la Historia de Usuario 8. Eliminar Notificaciones.....	67
Tabla 24: Plan de prueba para la Historia de Usuario 9. Crear diccionario.....	67
Tabla 25: Plan de prueba para la Historia de Usuario 11. Ejecutar algoritmo de análisis.....	68
Tabla 26: Plan de prueba para la Historia de Usuario 12. Etiquetar grupos. ....	68
Tabla 27: Diseño de caso de prueba para la Historia de Usuario 12. Etiquetar grupos.....	69
Tabla 28: Plan de prueba para la Historia de Usuario 13. Enviar grupos.....	69
Tabla 29: Diseño de caso de prueba para la Historia de Usuario 13. Enviar grupos. ....	70
Tabla 30: Prueba de Aceptación para la HU Adicionar sesión. ....	70
Tabla 31: Prueba de Aceptación para la HU Eliminar sesión. ....	71
Tabla 32: Prueba de Aceptación para la HU Insertar tarea. ....	71
Tabla 33: Prueba de Aceptación para la HU Eliminar tarea. ....	72
Tabla 34: Prueba de Aceptación para la HU Crear notificación. ....	72
Tabla 35: Prueba de Aceptación para la HU Eliminar notificación. ....	72
Tabla 36: Prueba de Aceptación para la HU Crear diccionario. ....	73
Tabla 37: Prueba de Aceptación para la HU Ampliar el ámbito de análisis.....	73
Tabla 38: Prueba de Aceptación para la HU Ejecutar agrupamiento. ....	73
Tabla 39: Prueba de Aceptación para la HU Etiquetar grupos. ....	74
Tabla 40: Prueba de Aceptación para la HU Enviar resultados.....	74
Tabla 41: Historia de usuario número 4. Modificar tareas. ....	48
Tabla 42: Historia de usuario número 7. Enviar notificaciones.....	49
Tabla 43: Plan de prueba para la Historia de Usuario 4. Modificar Tarea. ....	50
Tabla 44: Diseño de caso de prueba 4. Modificar Tarea. ....	50
Tabla 45: Plan de prueba para la Historia de Usuario 7: Enviar Notificaciones. ....	51
Tabla 46: Diseño de caso de prueba 7. Enviar Notificaciones. ....	51

Tabla 47: Prueba de Aceptación para la HU Modificar tareas. ....	52
Tabla 48: Prueba de Aceptación para la HU Enviar notificación. ....	52
Tabla 49: Plan de prueba para la Historia de Usuario 10. Ampliar el ámbito de análisis. ....	53
Tabla 50: Diseño de caso de prueba 10. Ampliar el ámbito de análisis. ....	53

## INTRODUCCIÓN

Con el fin de obtener buenos resultados en cuanto a la automatización de los sistemas, muchas empresas, compañías e instalaciones aplican programas de Supervisión, Control y Adquisición de Datos (SCADA), los mismos están especialmente diseñados para funcionar sobre ordenadores. Proporcionan comunicación con los dispositivos de campo (controladores autónomos) y controlan el proceso de forma automática desde la pantalla del ordenador. Estos sistemas proveen de toda la información que se genera en el proceso productivo a diversos usuarios y actualmente son empleados en una gran variedad de aplicaciones de diversas industrias: pesadas, ligeras o de bienes. [1]

El ambiente de producción en los que se despliegan los sistemas SCADA genera un gran volumen de datos difíciles de analizar, tanto para los operadores como para los directivos. Además, las interfaces hombre máquina con que cuentan estos sistemas de supervisión y control se limitan a la representación de una pequeña parte del total de los valores adquiridos, los cuales responden a necesidades puntuales de las operaciones supervisadas desde las consolas. Por ello, existe un alto grado de inutilidad de los datos almacenados, pues los mismos pudieran ser empleados para generar conocimientos relacionados con el proceso de producción, mediante herramientas especializadas para esta finalidad.

En el Centro de Informática Industrial (CEDIN), se desarrollan sistemas SCADA. Después de varios años de experiencia, con un producto actualmente desplegado en varias instalaciones de la empresa Petróleos de Venezuela S.A. (PDVSA), en la República Bolivariana de Venezuela; se persigue la intención de desarrollar otros sistemas similares y aumentar el valor agregado de estas nuevas soluciones. Una de las herramientas creadas para ello es un Almacén de Datos, para brindar la posibilidad de realizar análisis de datos históricos, mediante el uso de técnicas y aplicaciones, las cuales ayudan a trazar estrategias, realizar planificaciones, estimaciones futuras y tomas de decisiones sobre el proceso de negocio que se lleve a cabo.

Uno de las finalidades que persigue el análisis de históricos en el SCADA es conocer el estado de los equipos industriales dentro del proceso de producción, así como su desempeño en el mismo. También se han identificado otras áreas en las cuales se puede aplicar: la gestión del mantenimiento operacional de los equipos industriales, conocer el desempeño de la maquinaria a partir de la detección de fallas o el reconocimiento de estas, así como la posible relación existente entre el funcionamiento de los diferentes equipos.

El paso previo a estas tareas es la definición de los activos sobre los cuales se desea realizar el análisis. Actualmente esta información se maneja como un dato empírico en el proceso de producción por los operadores y especialistas de las plantas de las industrias, y al menos en los sistemas SCADA que se desarrollan en el CEDIN no se manejan dichos valores.

No obstante, el Almacén de Datos, mediante las dimensiones que define, permite proporcionar niveles de detalles en los análisis. Dentro de las dimensiones diseñadas, para la granularidad del proceso de generación de conocimiento, se tiene una para los equipos industriales con los que se integran las variables y dispositivos del SCADA. Para poblar el Almacén de Datos se efectúa el proceso ETL de los datos almacenados en la Base de Datos Históricas y en la Base de Datos de Configuración.

El sistema SCADA se configura para que interactúe con dispositivos de campo. Empíricamente se conoce que estos últimos guardan relación con sus equipos correspondientes. Sin embargo la dimensión Equipos Industriales no se puebla debido a que no se tiene una fuente de datos bien definida con esta información.

Tomando como ejemplo la experiencia de PDVSA, para la configuración de las variables en el SCADA se define una Norma por cada instalación de la industria. De esta manera se brindan especificaciones sobre cómo cada variable está relacionada con el equipo correspondiente. Estos datos se reflejan como una descripción dentro de la configuración de cada variable del SCADA, por tanto para saber al menos el tipo de equipo habría que consultar los datos asignados a cada una; tarea que puede demandar mucho tiempo si se tienen en cuenta que en muchas ocasiones la cantidad de variables en las instalaciones sobrepasa la cifra de cinco mil.

Por otra parte, estos datos encontrados en configuración son introducidos por los usuarios y se pueden apreciar errores en la escritura, abreviaturas y símbolos que varían según la Norma o el propio usuario. Siendo un impedimento para el uso de las herramientas ETL, para poblar la dimensión Equipo Industrial, debido a la incalculable cantidad de combinaciones en las que pueden aparecer los datos.

Por la complejidad que tiene obtener la información requerida para completar el proceso ETL, se pudiera emplear herramientas especializadas que utilicen algoritmos de Inteligencia Artificial. Pero para ello se requiere tener conocimientos en esta área de la informática y generalmente los analistas de datos estudian matemáticas o estadísticas, los operadores se limitan al uso de la interfaz hombre-máquina del SCADA, los directivos no son muy diestros en el tema; en fin, es necesario designar a una persona para ello.

Aún, al llegar a dominar alguna herramienta con estas características, no se garantiza que el resultado que se proporciona pueda ser exportado hacia una fuente de datos alternativa, por lo cual la información obtenida se debería introducir manualmente en el proceso ETL. Siendo una tarea que puede conllevar mucho tiempo y esfuerzo para realizarse, pues se debe tener en cuenta que en muchas ocasiones los SCADA tienen configuradas más de cinco mil variables, y es indispensable relacionar cada una con su Equipo Industrial correspondiente.

Por esta razón surge como **problema científico**: ¿Cómo obtener un mayor conocimiento de los equipos que intervienen en el proceso de producción de las industrias a partir de la configuración del SCADA, para poblar la dimensión Equipo Industrial del Almacén de Datos?

Teniendo en cuenta el problema anteriormente descrito la investigación se plantea como **objeto de estudio**: técnicas de Minería de Datos para el descubrimiento de información sobre bases de datos.

Pero como en esta área convergen muchos estudios se plantea como **campo de acción**: algoritmos de agrupamiento de datos sobre las Bases de Datos. Por tanto el **objetivo general** de esta investigación es: Desarrollar una herramienta que permita automatizar el análisis de las Bases de Datos de Configuración de SCADA para proporcionar información relevante requerida para poblar la dimensión Equipo Industrial del Almacén de Datos.

Para dar cumplimiento al objetivo definido se plantearon las siguientes **tareas de investigación**:

1. Búsqueda y estudio del estado del arte de los almacenes de datos y las herramientas que se emplean en el desarrollo del mismo.
2. Estudio de herramientas y tecnologías para realizar Minería de Datos.
3. Investigación sobre los algoritmos de Inteligencia Artificial basados en agrupamiento de datos.
4. Estudio de la estructura de almacenamiento de la Base de Datos de Configuración.
5. Estudio sobre el desarrollo de una herramienta de gestión de tareas.
6. Desarrollo de una herramienta que realice tareas de análisis basado en el modelo cliente – servidor.
7. Pruebas de funcionamiento de la aplicación obtenida.
8. Corrección de no conformidades.

Como **idea a defender** se plantea que: Un analista de datos puede obtener información sobre la relación que existe entre las variables configuradas en el SCADA mediante el uso de algoritmos y técnicas de Minería de Datos, entonces se puede minimizar el tiempo requerido para este proceso, automatizando el uso de estos algoritmos y técnicas.

Para llevar a cabo la investigación se hace uso de los siguientes **Métodos de trabajo científico**:

#### **Métodos teóricos:**

- **Analítico- sintético:** Se utilizó para seleccionar y resumir el amplio conocimiento reflejado en los materiales consultados sobre las técnicas y tareas de la Minería de Datos, a su vez cumplir el estudio de los aspectos definidos en el campo de acción y conformar el estado del arte.
- **Histórico-lógico:** Se utilizó para estudiar la evolución y desarrollo de las herramientas existentes para aplicar Minería de Datos con el fin de compararlas, conocer el estado actual de las mismas.
- **Inductivo - Deductivo:** Se utilizó para obtener conclusiones a partir del resultado de aplicar los diferentes algoritmos de asociación y agrupamiento para obtener asociaciones en los datos.
- 

#### **Método empírico:**

- **Entrevista:** Se utilizó en la realización de consultas a expertos con el fin de obtener información importante para los objetivos de la investigación.
- **Consulta de información en fuentes confiables:** para la elaboración del marco teórico de la investigación.

El presente documento está estructurado en los capítulos que a continuación se muestran:

**Capítulo 1 Fundamentación Teórica:** En este capítulo se exponen los principales conceptos relacionados con el tema, se realiza un estudio sobre las herramientas usadas para la fase de Minería de Datos dentro del proceso de extracción de conocimiento. Se definen las tareas, técnicas y algoritmos que engloba la minería.

**Capítulo 2 Diseño de la solución:** En este capítulo se explica el proceso ETL dentro de la solución del negocio, se plantean las funcionalidades del sistema y se define la arquitectura. Se exponen los principales productos de trabajo como parte de la técnica de modelado ágil de la metodología AUP. Entre las cuales se encuentran el diagrama de clases y de colaboración. También se definen las historias de usuarios y las tarjetas CRC.

**Capítulo 3 Construcción y Validación:** En este capítulo se selecciona la tecnología y se describe la metodología de desarrollo empleada, con el fin de guiar todo el proceso de desarrollo de la herramienta de análisis de tareas. Se describe el algoritmo implementado para dar solución al problema. Se validan las funcionalidades y se hacen las pruebas pertinentes para comprobar los resultados obtenidos.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

## Introducción

El objetivo de este capítulo es tratar los aspectos y conceptos importantes que ayudan a comprender el desarrollo de la investigación. Se definen las principales herramientas, tareas y técnicas usadas para la fase de Minería de Datos dentro del proceso de extracción de conocimiento, con el fin de seleccionar las más adecuadas para dar solución al problema. Además se describe de forma general los principales algoritmos que comprende la Minería de Datos.

### 1.1 Inteligencia de Negocio

La Inteligencia de Negocio (BI, por sus siglas en inglés), es la combinación de tecnología, herramientas y procesos que permiten transformar los datos almacenados en información, esta información en conocimiento y este conocimiento aplicarlo a un plan o una estrategia comercial. La Inteligencia de Negocio debe ser parte de la estrategia empresarial, esta permite optimizar la utilización de recursos, monitorear el cumplimiento de los objetivos de la empresa y la capacidad de tomar buenas decisiones para así obtener mejores resultados. [2]

Las herramientas de inteligencia se basan en la utilización de un sistema de información que se forma con distintos datos extraídos relacionados con la producción, la empresa, o sus ámbitos. La vida o el período de éxito de un software de Inteligencia de Negocio depende únicamente del éxito de su uso en beneficio de la empresa. Si esta empresa es capaz de incrementar su nivel financiero, administrativo y sus decisiones, mejora la actuación de la empresa y el software de Inteligencia de Negocio seguirá presente mucho tiempo, en caso contrario será sustituido por otro que aporte resultados más precisos. [3]

Entre las principales causas que conllevan a aplicar Inteligencia de Negocio en una empresa están: [2]

- Se tienen datos pero se carece de información – Es importante almacenar los datos referentes al negocio en aplicaciones, sistemas financieros o fuentes de datos. Sin embargo, si se quiere que la empresa tenga mayor ventaja sobre la competencia esta gestión no es suficiente. Se necesita profundizar en el nivel de conocimiento de estos datos para así, tener la capacidad de encontrar



patrones de comportamiento, monitorear, rastrear, entender, administrar y contestar aquellas interrogantes que permitan maximizar el rendimiento de la empresa.

- Fragmentación – Poseen aplicaciones independientes a través de todos los departamentos pero se carece de una visión global de la empresa.
- Manipulación manual- La necesidad de generar análisis de negocios e informes conlleva a utilizar herramientas de Inteligencia de Negocio y/o de reportes que no son las más confiables. Esta práctica trae consigo la exportación de datos a distintas herramientas que resultan en un proceso lento, costoso, duplicación de trabajo, poca confiabilidad en los informes, propenso a errores y sujetos a la interpretación individual.
- Poca agilidad – Debido a la carencia de información, la fragmentación y la manipulación manual el negocio se mantiene en un nivel de rendimiento bajo. Se necesita de una herramienta lo suficientemente ágil que se ajuste a las necesidades del negocio.

#### Beneficios de la Inteligencia de Negocio

Dentro del marco de beneficios que representa una solución de Inteligencia de Negocio se puede mencionar que esta permite: [2]

- Manejar el crecimiento – El reto para las empresas es evolucionar, es crecer y esto significa “cambio”. Que tan ágiles son mis procesos para enfrentar los cambios y las necesidades puntuales de la empresa.
- Control de costos – El manejo de costos es el detonador que fuerza muchas empresas a considerar una solución de Inteligencia de Negocio, para tener la capacidad de medir gastos y ver esto a un nivel de detalle que identifique la línea de negocio, producto, centro de costo, entre otras.
- Entender mejor a los clientes – Las empresas almacenan toneladas de información valiosa relacionada a sus clientes. El reto es transformar esta información en conocimiento y este conocimiento dirigido a una gestión comercial que represente algún tipo de ganancia para la empresa.
- Indicadores de gestión – Los indicadores de desempeño permiten representar medidas enfocadas al desempeño organizacional con la capacidad de representar la estrategia

organizacional en objetivos, métricas, iniciativas y tareas dirigidas a un grupo y/o individuos en la organización.

Principales productos de Inteligencia de Negocio:

- Sistema para la Toma de Decisiones (DSS).
- Cuadro de Mandos Integrales (CMI).

Los componentes fundamentales son:

- Almacenes de Datos
- Mercado de Datos

## **1.2 Proceso de extracción de conocimiento**

El descubrimiento de conocimiento de las bases de datos (KDD, por sus siglas en inglés) es una contestación a los enormes volúmenes de datos que son reunidos y almacenados en las bases de datos operacionales y científicas. Dado el aumento inevitable de la información en muchos dominios, como proceso de mejora en la tecnología de información (IT, por sus siglas en inglés), se adoptan de manera extendida los procesos de supervisión y control. EL KDD aporta influencias en la IT buscando información profundamente oculta que puede convertirse en conocimiento para tomar decisiones estratégicas y responder preguntas de investigación fundamentales. [4]

KDD abarca los principios y las técnicas de la estadística, máquina de aprendizaje, el reconocimiento de patrones, la búsqueda numérica, y la visualización científica para acomodar los nuevos tipos de datos y volúmenes de los datos generados a través de la IT [4]. El KDD es el proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos. (Fallad et al., 1996). El proceso KDD consta de una secuencia iterativa de etapas o fases. Dentro de las fases se encuentran: Integración y recopilación, Selección, limpieza y transformación, Minería de Datos, Evaluación e interpretación, y por último Difusión y uso.

### 1.2.1 Fases del KDD

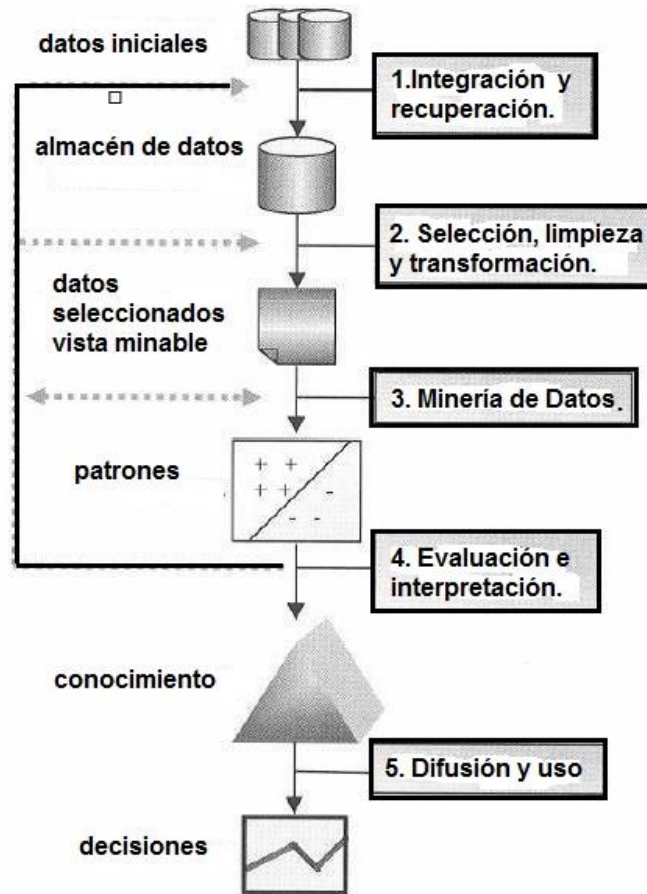


Figura 1. Fases del proceso de extracción de conocimiento en bases de datos.

### 1.2.2 Las metas del KDD

- Procesar automáticamente grandes cantidades de datos crudos.
- Identificar los patrones más significativos y relevantes.
- Presentarlos como conocimiento apropiado para satisfacer las metas del usuario.

El KDD es bien conocido a través del término más popular “Minería de Datos”. Sin embargo, la misma es sólo un componente (aunque un componente central) del largo proceso de KDD. La Minería de Datos involucra destilar los datos en información o hechos sobre el dominio descrito por la base de datos. KDD es el proceso de nivel-superior de obtener la información a través de la Minería de Datos y convertir esta información en el conocimiento (las ideas y creencias sobre el dominio) a través de la interpretación de información e integración con el conocimiento existente. [4]

### 1.3 Minería de Datos. Características y objetivos

Se define a la Minería de Datos como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Entre los principales métodos de la Minería de Datos figuran trabajar con grandes volúmenes de datos, procedentes mayoritariamente de los sistemas de información y usar técnicas adecuadas para analizar los mismos.

La Minería de Datos es una disciplina que combina técnicas de Inteligencia Artificial, Aprendizaje Computacional, Probabilidades, Estadísticas, y Bases de Datos para extraer información y conocimientos a partir de grandes cantidades de datos. La Minería de Datos permite a empresas/investigadores hacer explícito el conocimiento y utilizarlo en procesos de toma de decisiones. [5]

En los últimos años, la Minería de Datos ha experimentado un auge como soporte para la gestión de la información y el conocimiento, como alternativa a la modelación matemática. La Minería de Datos abarca un terreno muy amplio, no es solamente aplicar un algoritmo existente a un conjunto de datos. Las herramientas existentes actualmente incluyen mecanismos para la preparación de los datos, su visualización y la interpretación de los resultados. [5]

La Minería de Datos presenta dos modelos: uno predictivo para los problemas de aprendizaje supervisado en donde se estiman valores futuros o desconocidos y otro modelo descriptivo para los problemas de aprendizaje no supervisado, donde se identifican patrones que explican o resumen los datos, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos datos. También existen tareas que se dividen en predictivas y descriptivas, algunas de las tareas de Minería de Datos que producen modelos predictivos son la clasificación y la regresión, mientras que el agrupamiento, las reglas de asociación y el análisis correlacional dan lugar a modelos descriptivos.

#### 1.3.1 Técnicas de Minería de Datos

Dado que la Minería de Datos es un campo muy interdisciplinario, existen diferentes paradigmas detrás de las técnicas utilizadas para esta fase: [6]

Técnicas basadas en **redes neuronales artificiales**: se trata de técnicas que aprenden un modelo mediante el entrenamiento de los pesos que conectan un conjunto de nodos o neuronas. La topología de la red y los

pesos de las conexiones determinan el patrón aprendido. Existen innumerables variantes de organización: perceptrón simple, redes multicapa, redes de base radial, redes de Kohonen, etc.

Técnicas basadas en **árboles de decisión y sistemas de aprendizaje de reglas**: son técnicas que, además de su representación en forma de reglas, se basan en dos tipos de algoritmos: los algoritmos denominados "divide y vencerás", como el ID3/C4.5 o el CART, y los algoritmos denominados "separa y vencerás", como el CN2.

Técnicas basadas en **casos, en densidad o distancia**: son métodos que se basan en el cálculo de la distancia de cada elemento con respecto al resto ellos, ya sea directamente, como los vecinos más próximos (los casos más similares), de una manera más sofisticada, mediante la estimación de funciones de densidad. Además de los vecinos más próximos, algunos algoritmos muy conocidos son los jerárquicos, como Two-step o COBWEB, y los no jerárquicos, como K medias.

Técnicas **estocásticas y difusas**: en este marco se incluyen la mayoría de las técnicas que, junto a las redes neuronales, forman lo que se denomina computación flexible. Son técnicas en las que o bien los componentes aleatorios son fundamentales, como el recocido simulado, los métodos evolutivos y genéticos, o bien al utilizar funciones de pertenencia difusas.

Técnicas basadas en **núcleo y máquinas de soporte vectorial**: se trata de técnicas que intentan maximizar el margen entre los grupos o las clases formadas. Para ello se basan en unas transformaciones que pueden aumentar la dimensionalidad. Estas transformaciones se llaman núcleos. Existen muchísimas variantes, dependiendo del núcleo utilizado y de la manera de trabajar con el margen.

Técnicas **algebraicas y estadísticas**: se basan, generalmente, en expresar modelos y patrones mediante fórmulas algebraicas, funciones lineales, funciones no lineales, distribuciones o valores agregados estadísticos tales como medias, varianzas, correlaciones, etc. Frecuentemente, estas técnicas, cuando obtienen un patrón, lo hacen a partir de un modelo ya predeterminado del cual, se estiman unos coeficientes o parámetros, de ahí el nombre de técnicas paramétricas. Algunos de los algoritmos más conocidos dentro de este grupo de técnicas son la regresión lineal (global o local), la regresión logarítmica y la regresión logística. Los discriminantes lineales y no lineales, basados en funciones predefinidas, es decir discriminantes paramétricos, entran dentro de esta categoría.

No obstante, aunque el término "no paramétrico" se utiliza para englobar gran parte de técnicas provenientes del aprendizaje automático, como son las redes neuronales, también existen muchas técnicas de modelización estadística no paramétrica.

Técnicas basadas en **conteos de frecuencias y tablas de contingencia**: estas técnicas se basan en contar la frecuencia en la que dos o más sucesos se presenten conjuntamente. Cuando el conjunto de sucesos posibles es muy grande, existen algoritmos que van comenzando por pares de sucesos e incrementando los

conjuntos sólo en aquellos casos que las frecuencias conjuntas superen un cierto umbral. Ejemplos de estos algoritmos son el algoritmo "Apriori" y similares.

### **1.3.2 Tareas de Minería de Datos**

Dentro de la Minería de Datos se distinguen tipos de tareas, cada una de las cuales puede considerarse como un tipo de problema a ser resuelto por un algoritmo de Minería de Datos. Esto significa que cada tarea tiene sus propios requisitos, y que el tipo de información obtenida con una tarea puede diferir mucho de la obtenida con otra. [6]

#### **Clasificación**

La clasificación es quizás la tarea más utilizada. En ella, cada instancia (o registro de la base de datos) pertenece a una clase, la cual se indica mediante el valor de un atributo denominado la clase de la instancia. Este atributo puede tomar diferentes valores discretos, cada uno de los cuales corresponde a una clase. El resto de los atributos de la instancia (los relevantes a la clase) se utilizan para predecir la clase. El objetivo es predecir la clase de nuevas instancias de las que se desconoce la clase. En concreto, el objetivo del algoritmo es maximizar la razón de precisión de la clasificación de las nuevas instancias, la cual se calcula como el cociente entre las predicciones correctas y el número total de predicciones (correctas e incorrectas).

#### **Regresión**

La regresión es también una tarea predictiva que consiste en aprender una función real que asigna a cada instancia un valor real. Ésta es la principal diferencia respecto a la clasificación; el valor a predecir es numérico. El objetivo en este caso es minimizar el error (generalmente el error cuadrático medio) entre el valor predicho y el valor real.

#### **Agrupamiento**

El agrupamiento es la tarea descriptiva por excelencia y consiste en obtener grupos "naturales" a partir de los datos. Se habla de grupos y no de clases, porque, a diferencia de la clasificación, en lugar de analizar datos etiquetados con una clase, los analiza para generar esta etiqueta. Los datos son agrupados basándose en el principio de maximizar la similitud entre los elementos de un grupo minimizando la similitud entre los distintos grupos. Es decir, se forman grupos tales que los objetos de un mismo grupo son muy similares entre sí y, al mismo tiempo, son muy diferentes a los objetos de otro grupo.

Al agrupamiento también se le suele llamar segmentación, debido a que parte o segmenta los datos en grupos que pueden ser o no disjuntos. El agrupamiento está muy relacionado con la sumarización, que algunos autores consideran una tarea en sí misma, en la que cada grupo formado se considera como un resumen de los elementos que lo forman para así describir de una manera concisa los datos.

### **Análisis correlacional**

Las correlaciones son una tarea descriptiva que se usa para examinar el grado de similitud de los valores de dos variables numéricas. Una fórmula estándar para medir la correlación lineal es el coeficiente de correlación  $r$ , el cual es un valor real comprendido entre -1 y 1. Si  $r$  es 1 las variables están perfectamente correlacionadas, si  $r$  es -1 están correlacionadas negativamente, mientras que si es 0 no hay correlación. Esto quiere decir que cuando  $r$  es positivo, las variables tienen un comportamiento similar (ambas crecen o decrecen al mismo tiempo) y cuando  $r$  es negativo si una variable crece la otra decrece. El análisis de correlaciones, sobre todo de las negativas, puede ser muy útil para establecer reglas de elementos correlacionados.

### **Reglas de asociación**

Las reglas de asociación son también una tarea descriptiva, muy similar a las correlaciones, que tiene como objetivo identificar relaciones no explícitas entre atributos categóricos. Pueden conformarse de muchas formas, aunque su expresión más común es del estilo "si el atributo X toma el valor d entonces el atributo Y toma el valor b". Las reglas de asociación no implican una relación causa-efecto, es decir, puede no existir una causa para que los datos estén asociados. Las reglas se evalúan usando dos parámetros: precisión y soporte (cobertura).

## **1.4 Algoritmos de Minería de Datos**

C4.5:

El algoritmo C4.5 es la extensión del algoritmo ID3 para la generación de árboles de decisión. El C4.5 visita recursivamente cada nodo de decisión, seleccionando un particionador óptimo, hasta que no sean posibles más particiones. El algoritmo C4.5 usa el concepto de ganancia de información o reducción de la entropía para seleccionar la partición óptima. C4.5 utiliza el concepto de entropía suponiendo que se tiene una partición candidata  $S$ , la cual particiona el conjunto de datos de entrenamiento  $T$  en varios subconjuntos  $T_1, T_2, \dots, T_k$ . El requerimiento informacional puede ser calculado como la suma de los pesos de la entropía para un subconjunto individual.

De la forma:  $H_s(T) = \sum_{i=1}^k P_i H_s(T_i)$

Donde  $P_i$  representa la proporción de registros en el subconjunto  $i$ .

Entonces se puede definir la ganancia de información a obtener como  $\text{ganancia}(s) = H(T) - H_s(T)$ , o sea el incremento de información producida por el particionamiento de los datos de

entrenamiento  $T$  de acuerdo con el particionador candidato  $S$ . En cada nodo de decisión,  $C4.5$  selecciona la partición óptima que tenga la mayor ganancia ( $S$ ). [7]

Regresión lineal simple:

La regresión lineal es un método simple pero frecuentemente utilizado para la tarea de regresión, que dado dos variables  $X$  y  $Y$  determina si existe una relación (aproximada) lineal entre  $X$  y  $Y$ . El problema matemático es encontrar  $a, b \in \mathbb{R}$  tal que:  $y_i = ax_i + b + e_i$  para  $i = 1, 2, \dots, n$  de modo que los  $e_i$  sean los más pequeños posible. Concretamente se usa el criterio de los mínimos cuadrados, es decir se quiere que la suma  $\sum_{i=1}^n e_i^2$  sea mínima. [8]

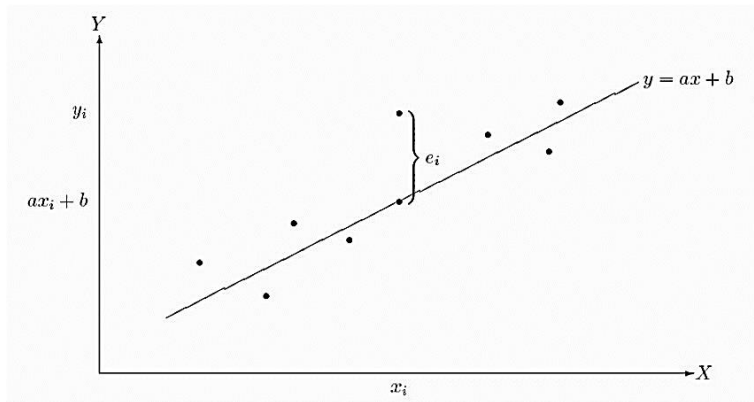


Figura 2. Representación gráfica de regresión lineal.

K-means:

Es un algoritmo no supervisado de agrupamiento que toma como parámetro de entrada  $k$ , el número de grupos que se desea encontrar, y parte un conjunto de  $n$  objetos en  $k$  grupos, tal que la similitud resultante dentro de un grupo es alta, pero la similitud con otros grupos es baja.

Este algoritmo busca una partición óptima de los datos minimizando el criterio de la suma del error cuadrado con un procedimiento iterativo de optimización, el cual pertenece a la categoría de algoritmos hill-climbing. [9]

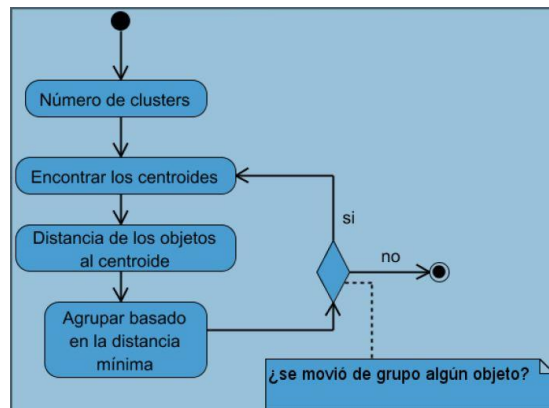




Figura 3. Algoritmo K-means.

#### COBWEB:

El algoritmo COBWEB presenta la noción de agrupamiento conceptual, se utiliza para justificar la necesidad de un agrupamiento cualitativo frente al agrupamiento cuantitativo, basado en la vecindad entre los elementos de la población. COBWEB forma los conceptos por agrupación de ejemplos con atributos similares. Representa los grupos como una distribución de probabilidad sobre el espacio de los valores de los atributos, generando un árbol de clasificación jerárquica en el que los nodos intermedios definen subconceptos. El objetivo de COBWEB es hallar un conjunto de clases o grupos (subconjuntos de ejemplos) que maximice la utilidad de la categoría (partición del conjunto de ejemplos cuyos miembros son clases). [10]

#### Apriori:

Principal algoritmo de las reglas de asociaciones, su funcionamiento se basa en la búsqueda de los conjuntos de elementos con determinado soporte y en el principio básico que plantea que cualquier subconjunto de un conjunto de elementos frecuentes debe ser frecuente.

A priori: [11]

-Genera los conjuntos frecuentes: si ellos tienen el primer k-1 artículo en común (en orden alfabético). Si cualquiera de estos subconjuntos no son frecuentes, s no puede ser frecuente y es entonces depurado.

-Seguidamente genera las reglas de asociación: primero genera todos los subconjuntos de s. Entonces permite a ss representar los subconjuntos de s no vacíos. Considera la regla de asociación R:  $ss \rightarrow (s-ss)$ , donde (s-ss) indica el conjunto s sin ss. Genera (la salida) R si R cumple el requerimiento mínimo de confianza y lo hace para cada subconjunto ss de s.

Para evaluar el potencial de las reglas se suele usar la proporción de confianza, esta se define:

$$\text{Proporción de confianza} = 1 - \min\left(\frac{p(y|x)}{p(y)}, \frac{p(y)}{p(y|x)}\right)$$

### 1.5 Base de Datos, Almacenes de Datos y Vista Minable

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Es un sistema de archivos electrónico [12]. Las bases de datos son el soporte del sistema de información de las organizaciones, diseñadas para dar soporte (eficiente) a las funciones básicas de la organización.

Existen diversos dominios donde se almacenan grandes volúmenes de información en bases de datos centralizadas y distribuidas. [30]

Base de Datos Distribuidas (BDD): Es un conjunto de múltiples bases de datos lógicamente relacionadas, las cuales se encuentran distribuidas en diferentes espacios lógicos e interconectados por una red de comunicaciones. Dichas BDD tienen la capacidad de realizar procesamiento autónomo, esto permite realizar operaciones locales o distribuidas. Un sistema de Bases de Datos Distribuidas (SBDD) es un sistema en el cual múltiples sitios de bases de datos están ligados por un sistema de comunicaciones de tal forma que, un usuario en cualquier sitio puede acceder a los datos en cualquier parte de la red exactamente como si estos fueran accedidos de forma local. [34]

Base de Datos Centralizadas: Una base de datos centralizada es una base de datos almacenada en su totalidad en un solo lugar físico, es decir, almacenada en una sola máquina, en donde los usuarios trabajan en terminales que solo muestran resultados. Los sistemas de bases de datos centralizadas son aquellos que se ejecutan en un único sistema informático sin interactuar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos mono usuarios ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas.

Un Almacén de Datos es una colección de datos orientado a temas o materias, integrado, no volátil y variable en el tiempo, que será utilizado fundamentalmente en el proceso de toma de decisiones.

Soporta decisiones de administración (donde el término no volátil significa que una vez que los datos han sido insertados, no pueden ser cambiados). Los almacenes de datos surgieron por dos razones: la necesidad de proporcionar una fuente única de datos limpia y consistente para propósitos de apoyo para la toma de decisiones; y por la necesidad de hacerlo sin afectar a los sistemas operacionales. [12]

La tecnología de los almacenes de datos (del inglés, Data Warehouse), se encuadra dentro de la línea de evolución de las bases de datos hacia una mayor funcionalidad e inteligencia. El uso de los almacenes de datos proporciona las siguientes ventajas: transforman los datos orientados a las aplicaciones en información orientada a la toma de decisiones.

Permiten un análisis inmediato de los resultados de compras y ventas, agilidad en el control de inventarios. Tienen la capacidad de analizar y explorar las diferentes áreas de trabajo. Provee una relación total con el cliente. Facilita la gestión y el análisis de los recursos. Conexiona departamentos empresariales (que antes formaban islas). Reaccionan rápidamente al cambio en el mercado. [29]

El Almacén de Datos se compone de tres partes:

- Adquisición:
  - Importación de datos procedentes de los sistemas OLTP (datos propios de la empresa, de fuentes externas) al Almacén de Datos.
  - El proceso de limpieza de los datos importados.
  - Actualizar el Almacén de Datos.
  
- Almacenamiento:
  - Se puede usar un SGBD relacional o SGBD multidimensional.
  - Lo más típico es usar un almacén central.
  
- Acceso:
  - Trata directamente con el usuario final, ofreciéndole herramientas de varios tipos:
    - Visualización de datos.
    - Análisis estadístico.
    - Generadores de informes.

Vista minable: Tras la recogida e integración de datos: el objetivo de la “Preparación de Datos” es obtener la “vista minable”, a partir de datos que pueden ser inadecuados, faltantes, erróneos, irrelevantes o dispersos. La vista minable es el conjunto de datos que incluyen todas las variables de interés para el problema concreto en el formato adecuado. La vista minable integra datos de diferentes fuentes, los limpia, selecciona y transforma, con el fin de prepararlos para aplicarles una técnica de Minería de Datos. [31]

## 1.6 Pentaho Data Integration

Un paso previo a la ejecución de los algoritmos de Minería de Datos es la realización de la extracción de los datos de los sistemas de almacenamiento. Pentaho Data Integration (PDI,

también llamado *Kettle*) es el componente responsable por el proceso ETL. PDI puede ser usado además para otros propósitos: [13]

- La migración de datos entre aplicaciones o base de datos.
- Exportar los datos de las bases de datos para los archivos sin interés.
- Cargar datos masivos en las bases de datos.
- La limpieza de los datos.
- Integración de aplicaciones.

PDI es fácil de usar. Cada proceso es creado con componentes gráficos donde se especifica cada operación sin necesidad de programar cómo hacerlo mediante un lenguaje; debido a esto, se podría decir que PDI es orientado a metadatos. PDI puede usarse como una aplicación autónoma, o puede usarse como parte de la colección de Pentaho más grande. Como una herramienta de ETL, es la herramienta de fuente abierta más popular disponible. PDI soporta una inmensa serie de formatos de entrada y salida, incluso archivos de texto, hojas de cálculos, y los artefactos de la base de datos comerciales y libres. Es más, la capacidad de transformación de PDI permite manipular los datos con muy pocas limitaciones. [13]

## ETL

Extraer: La primera parte del proceso ETL consiste en extraer los datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos fusionan datos provenientes de diferentes sistemas de origen. [14]

Transformar: La fase de transformación aplica una serie de reglas de negocios o funciones sobre los datos extraídos para convertirlos en datos que serán cargados. Algunas fuentes de datos requerirán alguna pequeña manipulación de los datos. No obstante en otros casos puede ser necesario aplicar algunas de las siguientes transformaciones: [14]

- Seleccionar solo ciertas columnas para su carga (ejemplo, que las columnas con valores nulos no se carguen).
- Traducir códigos (por ejemplo, si la fuente almacena una “H” para hombre y “M” para mujer pero el destino tiene que guardar “1” para hombre y “2” para mujer).
- Obtener nuevos valores de cálculos (por ejemplo,  $total\_venta=cantidad*precio$ ).
- Unir datos de múltiples fuentes (por ejemplo, búsquedas, combinaciones, etc.).

- Generación de campos claves en el destino.

Cargar: La fase de carga es el momento en el cual los datos de la fase anterior (transformación) son cargados en el sistema destino. Este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos, en otras se agrega la información ya existente. Todo depende del modelo y de los requerimientos del negocio. [14]

La fase de carga interactúa directamente con la base de datos de destino. Al realizar esta operación se aplicarán todas las restricciones y configuraciones que se hayan definido en esta (por ejemplo, valores únicos, integridad referencial, campos obligatorios) y si están bien definidos contribuyen a que se garantice la integridad de los datos en el proceso ETL. [14]

PDI Kettle consiste principalmente de las siguientes aplicaciones: [14]

- Spoon: Es el componente más utilizado. Es una herramienta gráfica que permite diseñar Jobs y Transformaciones ETL. Esta herramienta permite la conexión a diversos orígenes de datos y transformarlos para cargarlos dentro de su estructura del Data Warehouse.
- Kitchen: Es un programa que permite ejecutar los “jobs” diseñados en Spoon, permitiendo programarlos y ejecutarlos en modo batch.
- Pan: Permite ejecutar las transformaciones diseñadas en Spoon, con la ejecución desde la línea de comandos y ejecutarlos en modo batch. 1.7 WEKA

### **WEKA (Waikato Environment for Knowledge Analysis)**

Es una herramienta de aprendizaje automático y minería de datos, escrita en lenguaje Java, gratuita y desarrollada en la Universidad de Waikato. Es un entorno para experimentación de minería de datos que permite aplicar, analizar y evaluar las técnicas más relevantes de análisis de datos, principalmente las provenientes del aprendizaje automático, sobre cualquier conjunto de datos del usuario. Para ello únicamente se requiere que los datos a analizar se almacenen con un cierto formato, conocido como Archivo de Formato Atributo- Relación (ARFF, por sus siglas en inglés).

Está constituido por una serie de módulos con diferentes técnicas de: preprocesado, clasificación, agrupamiento, asociación y visualización. La licencia de WEKA es GPL, lo que significa que este programa es de libre distribución y difusión. Además, WEKA está programado en Java, es independiente de la arquitectura, funciona en cualquier plataforma sobre la que haya una máquina virtual Java disponible. [16]

WEKA. Ofrece 4 ambientes de trabajo:



Figura 4. Selector de interfaces de la herramienta WEKA.

Para poner en práctica las técnicas y algoritmos seleccionados se hizo uso de la interfaz Explorer. La misma permite llevar a cabo la ejecución de los algoritmos de análisis implementados sobre los ficheros de entrada, una ejecución independiente por cada prueba.

## 1.8 Conclusiones del Capítulo

Con el estudio del estado del arte se cumplieron los objetivos del presente capítulo. Se caracterizó la Minería de Datos dentro del proceso de extracción del conocimiento, definiéndose sus tareas y técnicas, así como sus algoritmos principales. Además se definieron los principales conceptos del proceso de negocio y se detalló la herramienta de Minería de Datos a utilizar. A partir del estudio del marco teórico conceptual se adquieren los conocimientos necesarios para pasar al diseño de la solución.

## **CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN**

### **Introducción**

En el presente capítulo se hace el diseño de la propuesta de solución para dar cumplimiento al problema a resolver. Se describen las funcionalidades que el sistema debe cumplir. Se explica el proceso ETL ajustado a la solución. Se define la arquitectura de la solución propuesta y se generan los siguientes artefactos: Diagrama de Clases, Diagrama de Colaboración, Tarjetas CRC e Historias de Usuario.

### **2.1 Propuesta de solución**

Con el Almacén de Datos se tiene un ambiente de Inteligencia de Negocio. Para ampliar el ámbito de explotación del mismo se debe poblar la dimensión Equipo Industrial. Esta tarea debe aplicarse desde las diferentes instalaciones donde se despliegue el SCADA o al menos se almacene la Base de Datos de Configuración del mismo. Para lograr una abstracción con el ambiente de producción en el proceso industrial, debido a la alta complejidad de las tareas que se quieren automatizar, y a la intención de que múltiples usuarios hagan uso de la misma de manera simultánea, en la presente investigación se propone una herramienta que centralice el proceso de análisis de los datos, proporcionando una mejor administración, mantenimiento y soporte del mismo.

Se toma como punto de partida, para ello, la particularidad que tienen las técnicas de Minería de Datos; pues son soluciones genéricas a diversos problemas. Por tanto la automatización de estas tareas no se limita al ambiente industrial, provee un espacio de trabajo que puede ser utilizado para otros análisis. Teniendo en cuenta lo anterior, la solución formaría parte del proceso ETL del almacén de datos, pero de forma desacoplada a este.

#### **El proceso ETL**

- Dentro del Proceso ETL el cliente identifica cuales son los datos que deben analizar para poblar la dimensión Equipo Industrial, luego realiza la extracción de los datos contenidos en la Base de Datos de Configuración y los importa en la herramienta de análisis.
- A partir de los datos obtenidos, la herramienta de análisis realiza la transformación de los mismos aplicando un algoritmo de Inteligencia Artificial, esto proporciona los resultados necesarios a utilizar para el resto del proceso.

- Una vez que el cliente obtiene estos resultados los puede insertar en la dimensión Equipo Industrial del almacén como parte de la etapa de carga en el ETL.

## 2.2 Modelo del Dominio

El modelo de dominio es una representación visual de los conceptos u objetos de los procesos de negocio o áreas funcionales para realizar una descripción que permita comprender la esencia de la actividad que realiza la entidad para la que se desarrolla el producto de software. [36]

A continuación se presenta el modelo conceptual correspondiente al Sistema de análisis de tareas, así como la descripción de los conceptos involucrados.

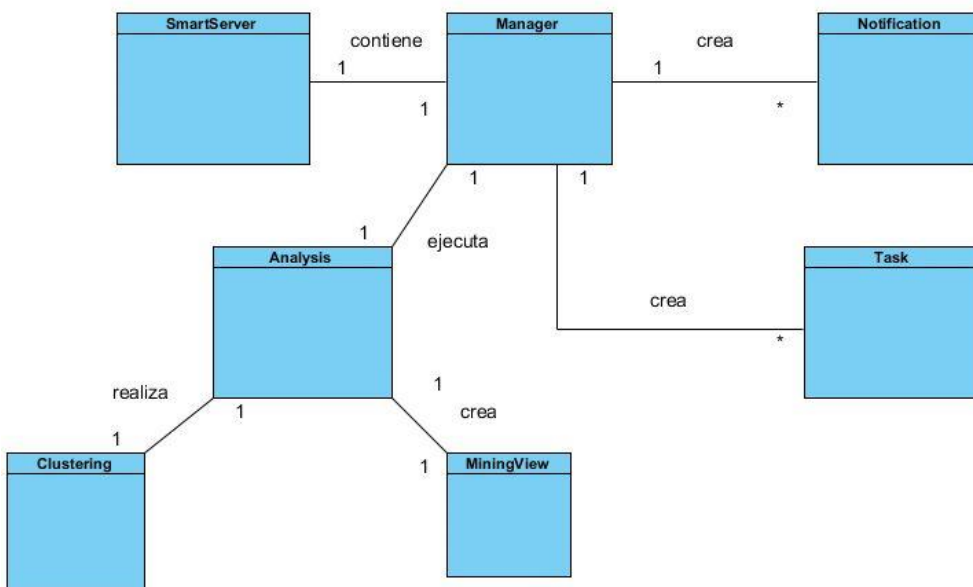


Figura 5. Diagrama de Modelo del Dominio.

Descripción del Modelo del Dominio para lograr un mejor entendimiento de la propuesta de solución

SmartServer representa el objeto con el cual el cliente establece una comunicación para enviar los datos a analizar.

Manager, representa el objeto controlador encargado de crear notificaciones y tareas. También establece cuando realiza el análisis de los datos recibidos.



Notification define las notificaciones que le van a llegar al cliente sobre el trabajo que se realice con las tareas que este envíe.

Task define la estructura de las tareas que envía el cliente.

Analysis es el objeto que se encarga de enviar el contenido de la tarea para llenar la vista minable, enviando la misma al algoritmo de Inteligencia Artificial.

MiningView representa el objeto que configura el diccionario y la matriz de los elementos, usando el contenido y las etiquetas definidas por el usuario. Proporciona el archivo de la vista minable en el servidor con el identificador otorgado por el algoritmo.

Clustering ejecuta al algoritmo de minería de datos y devuelve los grupos generados.

### 2.3 Funcionalidades del sistema

**Requisitos Funcionales:** Los requisitos funcionales describen las funcionalidades principales que debe cumplir el producto de software a desarrollar.

Requisitos funcionales	Descripción
<b>RF1: Adicionar sesión</b>	Permite recibir una nueva conexión de un cliente y guardar la misma en la lista de sesiones.
<b>RF2: Eliminar sesión</b>	Permite eliminar una conexión existente cuando recibe la petición de cerrar sesión.
<b>RF3: Insertar tareas</b>	Permite insertar una nueva tarea en la lista de tareas.
<b>RF4: Modificar tareas</b>	Permite modificar el contenido de una tarea existente en la lista de tareas.
<b>RF5 :Eliminar tareas</b>	Permite eliminar una tarea en la lista de tareas.
<b>RF6 :Crear notificaciones</b>	Permite crear una notificación con el estado de las tareas correspondientes al cliente.
<b>RF7: Enviar notificaciones</b>	Permite enviar al cliente correspondiente el estado de las tareas.
<b>RF8: Eliminar notificaciones</b>	Permite eliminar la notificación de la lista de notificaciones una vez que esta es enviada.
<b>RF9: Crear diccionario</b>	Permite crear un diccionario a partir de los elementos contenidos en la tarea enviada

	por el cliente.
<b>RF10. Ampliar el ámbito de análisis</b>	Permite crear una vista minable conformada por el diccionario y las etiquetas generadas por el cliente.
<b>RF11: Ejecutar algoritmo de análisis</b>	Permite el análisis de la vista minable mediante la ejecución del algoritmo.
<b>RF12: Etiquetar grupos</b>	Permite etiquetar los grupos obtenidos luego de aplicar el algoritmo de análisis.
<b>RF13: Enviar resultados</b>	Permite enviar un mensaje que contiene los grupos obtenidos, con sus respectivas etiquetas a los clientes.

Tabla 1: Requisitos funcionales del Servidor de análisis.

**Requisitos no funcionales:** Los requisitos no funcionales son los requerimientos que debe cumplir el entorno del producto de software a desarrollar para garantizar seguridad, usabilidad, rapidez y atracción.

- RNF1. Seguridad: El servicio controla que al usuario le llegue la información solicitada a partir de los datos que está proporcionando a través de la gestión de las sesiones.
- RNF2. Usabilidad: El acceso al servicio será de una manera fácil y rápida. El mismo está idealizado para que dada la entrada de uno o varios mensajes que contengan descripciones de puntos, entre otros datos, se genere la fuente de datos requerida para el posterior proceso de carga hacia el Almacén de Datos. Notificando durante todo el proceso de análisis el estado de las tareas en cuestión a los clientes.
- RNF3. Hardware: El ordenador para el uso de la herramienta de análisis de datos debe contar con no menos de 2 GB de memoria RAM. Procesador Intel Core i3 multinúcleo con velocidad igual o superior a 2.0 Ghz. Disco duro interno, de al menos 80 GB SATA 3.0 Gb/s 7200 RPM.
- RNF4 Portabilidad / Disponibilidad: El servidor permitirá su disposición independientemente del ambiente donde se desee desplegar.
- RNF5 Comunicación: El servicio garantizará la comunicación entre el cliente y el servidor, debido a que se quiere centralizar las funcionalidades de análisis, y el tiempo que estas requieren dependen de la cantidad de datos que procesen, se debe seleccionar una tecnología que permita notificar o hacerle saber al usuario cuando el proceso de descubrimiento haya terminado.

## 2.4 Historias de Usuario

Las historias de usuario son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes.

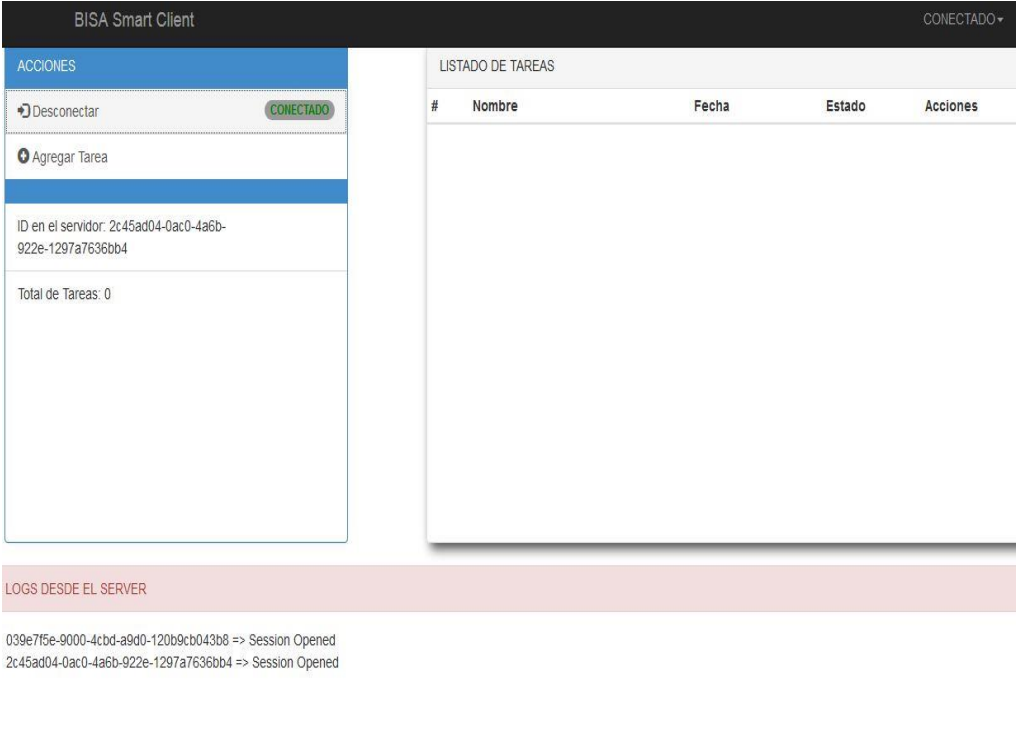
Historia de Usuario	
<b>Número: 1</b>	<b>Nombre del requisito: Adicionar sesión</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 1</b>
<b>Prioridad: Alta</b>	<b>Tiempo Estimado: 24h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 10h</b>
<b>Descripción:</b> Permite al servidor recibir una nueva conexión de un cliente, asignar un identificador al cliente conectado y guardar esta nueva conexión en la lista de sesiones.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	
 <p>The screenshot shows the BISA Smart Client interface. At the top, it says 'BISA Smart Client' and 'CONECTADO'. Below this, there are two main sections: 'ACCIONES' and 'LISTADO DE TAREAS'. The 'ACCIONES' section has buttons for 'Desconectar' (with a 'CONECTADO' indicator) and 'Agregar Tarea'. Below the actions, it displays the server ID: 'ID en el servidor: 2c45ad04-0ac0-4a6b-922e-1297a7636bb4' and 'Total de Tareas: 0'. The 'LISTADO DE TAREAS' section is a table with columns for '#', 'Nombre', 'Fecha', 'Estado', and 'Acciones', but it is currently empty. At the bottom, there is a 'LOGS DESDE EL SERVER' section with two log entries: '039e7f5e-9000-4cbd-a9d0-120b9cb043b8 =&gt; Session Opened' and '2c45ad04-0ac0-4a6b-922e-1297a7636bb4 =&gt; Session Opened'.</p>	

Tabla 2: Historia de usuario número 1. Adicionar sesión.

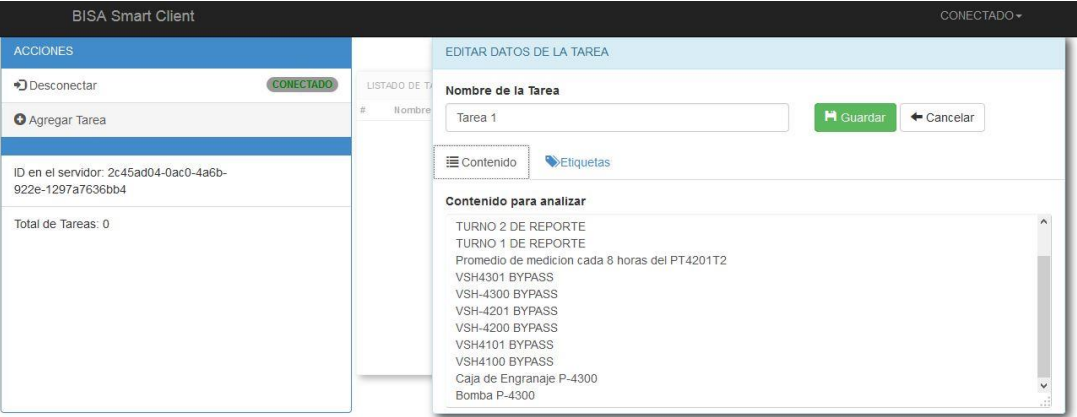
Historia de Usuario	
<b>Número: 3</b>	<b>Nombre del requisito: Insertar tareas</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 1</b>
<b>Prioridad: Alta</b>	<b>Tiempo Estimado:48h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 36h</b>
<b>Descripción:</b> Permite a la controladora insertar una nueva tarea en la lista de tareas una vez que el cliente la crea y la envía al servidor.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	
 <p>The screenshot shows the BISA Smart Client interface. On the left, there's a sidebar with 'ACCIONES' (Desconectar, Agregar Tarea) and a status bar showing 'ID en el servidor' and 'Total de Tareas: 0'. The main area displays 'EDITAR DATOS DE LA TAREA' with a form for 'Nombre de la Tarea' (Tarea 1) and 'Contenido' (a list of tasks). Buttons for 'Guardar' and 'Cancelar' are present. At the bottom, there's a 'LOGS DESDE EL SERVER' section with session logs.</p>	

Tabla 3: Historia de usuario número 3. Insertar tareas.

Historia de Usuario	
<b>Número: 5</b>	<b>Nombre del requisito: Eliminar tareas</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 1</b>
<b>Prioridad: Media</b>	<b>Tiempo Estimado: 4h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 3h</b>
<b>Descripción:</b> Permite a la controladora eliminar una tarea en la lista de tareas una vez que el cliente elimina una tarea previamente creada.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

LISTADO DE TAREAS				
#	Nombre	Fecha	Estado	Acciones
1	Tarea 1	2015/06/21	NUEVA	  

Tabla 4: Historia de usuario número 5. Eliminar tareas.

Historia de Usuario	
<b>Número:6</b>	<b>Nombre del requisito: Crear notificaciones</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 2</b>
<b>Prioridad: Alta</b>	<b>Tiempo Estimado: 24h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 23h</b>
<b>Descripción:</b> Permite a la controlador crear una notificación con el estado de las tareas correspondientes al cliente.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tabla 5: Historia de usuario número 6. Crear notificaciones.

Historia de Usuario	
<b>Número:8</b>	<b>Nombre del requisito: Eliminar notificaciones</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 2</b>
<b>Prioridad: Media</b>	<b>Tiempo Estimado: 4h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 3h</b>
<b>Descripción:</b> Permite eliminar la notificación de la lista de notificaciones una vez que esta es enviada.	
<b>Observaciones:</b>	

Prototipo de interfaz:

Tabla 6: Historia de usuario número 8. Eliminar notificaciones.

Historia de Usuario	
Número:9	Nombre del requisito: Crear diccionario
Programador: Deyanira Bonaparte Pérez.	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 72h
Riesgo en Desarrollo: Problemas técnicos y eléctricos.	Tiempo Real: 69h
<b>Descripción:</b> Permite crear un diccionario a partir de los elementos contenidos en la tarea enviada por el cliente.	
<b>Observaciones:</b>	
Prototipo de interfaz:	

Tabla 7: Historia de usuario número 9. Crear diccionario.

Historia de Usuario	
Número:10	Nombre del requisito: Ampliar el ámbito de análisis
Programador: Deyanira Bonaparte Pérez.	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 48h
Riesgo en Desarrollo: Problemas técnicos y eléctricos.	Tiempo Real: 48h
<b>Descripción:</b> Permite crear una vista minable conformada por el diccionario y las etiquetas generadas por el cliente.	
<b>Observaciones:</b>	
Prototipo de interfaz:	

EDITAR DATOS DE LA TAREA

**Nombre de la Tarea**

Contenido
Etiquetas

**Etiqueta**

Bomba

Tanque

Tabla 8: Historia de usuario número 10. Ampliar el ámbito de análisis.

Historia de Usuario	
<b>Número:11</b>	<b>Nombre del requisito: Ejecutar algoritmo de análisis</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 3</b>
<b>Prioridad: Alta</b>	<b>Tiempo Estimado:48h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 46h</b>
<b>Descripción:</b> Permite el análisis de la vista minable mediante la ejecución del algoritmo de agrupamiento.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

Tabla 9: Historia de usuario número 11. Ejecutar Algoritmo de análisis.

Historia de Usuario	
<b>Número:12</b>	<b>Nombre del requisito: Etiquetar grupos</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 3</b>
<b>Prioridad: Alta</b>	<b>Tiempo Estimado: 8h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 4h</b>

<b>Descripción:</b> Permite etiquetar los grupos obtenidos luego de aplicar el algoritmo de análisis.
<b>Observaciones:</b>
<b>Prototipo de interfaz:</b>
<pre>Instance 0 -&gt; Cluster 1 Instance 1 -&gt; Cluster 0 Instance 2 -&gt; Cluster 9 Instance 3 -&gt; Cluster 0 Instance 4 -&gt; Cluster 4 Instance 5 -&gt; Cluster 1 Instance 6 -&gt; Cluster 6 Instance 7 -&gt; Cluster 1 Instance 8 -&gt; Cluster 7 Instance 9 -&gt; Cluster 1 Instance 10 -&gt; Cluster 1 Instance 11 -&gt; Cluster 5 Instance 12 -&gt; Cluster 3 Instance 13 -&gt; Cluster 5 Instance 14 -&gt; Cluster 8 Instance 15 -&gt; Cluster 5 Instance 16 -&gt; Cluster 2</pre>

Tabla 10: Historia de usuario número 12. Etiquetar grupos.

Historia de Usuario	
<b>Número:13</b>	<b>Nombre del requisito: Enviar resultados</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 3</b>
<b>Prioridad: Alta</b>	<b>Tiempo Estimado:3h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 1h</b>
<b>Descripción:</b> Permite enviar un mensaje que contiene los grupos obtenidos, con sus respectivas etiquetas a los clientes.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	



RESULTADO DEL AGRUPAMIENTO	
Nombre de la Tarea	
Tiempo	Cluster: 1
TURNNO	Cluster: 8
TURNNO	Cluster: 0
TURNNO	Cluster: 0
Promedio	Cluster: 4
VSH4301	Cluster: 5
VSH-4300	Cluster: 6

← Cancelar

Tabla 11: Historia de usuario número 13. Enviar resultados.

## 2.5 Selección de la tecnología

### 2.5.1 Modelo Cliente-Servidor

Desde el punto de vista funcional, se puede definir la computación Cliente – Servidor como un modelo distribuido que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma. [25]

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. En otras palabras esta arquitectura es una extensión de la programación modular en la que la base fundamental es separar una gran pieza de software en módulos con el fin de hacer más fácil el desarrollo y mejorar su mantenimiento. [25]

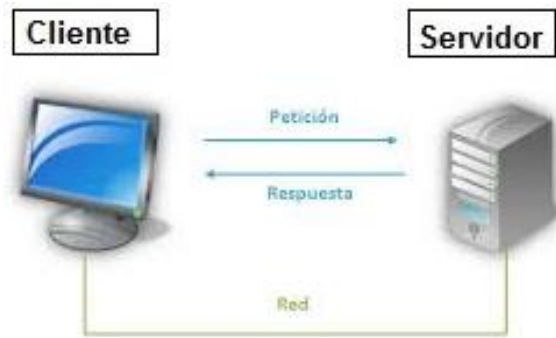


Figura 6. Modelo cliente-servidor.

## 2.5.2 WebSockets

WebSockets es un protocolo que permite la comunicación entre los clientes y los servidores y viceversa, sin un protocolo HTTP a la cabeza. Usa su propio protocolo que se define por el Grupo de Trabajo de Ingeniería de Internet (IETF, por sus siglas en inglés). También tiene una Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés) que puede usarse por las aplicaciones web para abrir y cerrar las conexiones, enviar y recibir los mensajes. A esto se le llama el API de WebSockets y se define en una especificación de la (W3C, por sus siglas en inglés). Con WebSockets se puede tener una plena comunicación bidireccional simultánea entre el servidor y el cliente con menos gastos que con el tradicional método basado en HTTP. Esto promete más rapidez, mayor escalabilidad y aplicaciones más robustas en tiempos reales en la web. Esto se traduce a mejoras en el desempeño de la red, sobre todo para aplicaciones que requieren de actualizaciones en tiempo real. [23]

### Funcionamiento de WebSockets

Antes de que el cliente y el servidor comiencen a enviar y recibir mensajes, es necesario establecer una conexión entre ellos primero. Esto se hace estableciendo un 'apretón de manos', donde el cliente manda una petición para conectar, y si el servidor quiere, mandará una contestación que acepta la conexión. La especificación del protocolo asegura que ambos, cliente y servidor basados en HTTP y WebSockets pueden operar en el bus de mensaje. [23]

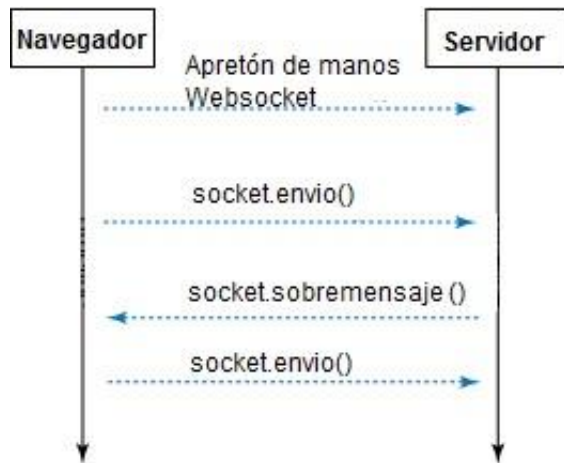


Figura 7. Ejemplo de comunicación mediante Websocket.

### 2.5.3 JSON

JSON, acrónimo de JavaScript Object Notation (notación de objetos java scripts) es un formato ligero para el intercambio de datos. Básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. JSON nació como una alternativa a XML. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías. [28]

Mediante este formato se pueden establecer protocolos de mensajes en el modelo cliente – servidor. Lo cual lo hace muy demandado a la hora de implementar soluciones soportadas con WebSockets. JSON es basado en texto, fácil de parsear. No define funciones, ni estructuras invisibles y tampoco es extensible [32]. JSON implementa funcionalidades que permiten transformar una cadena de caracteres en un objeto JavaScript y viceversa.

#### Sintaxis de JSON

La sintaxis de JSON es la mezcla de literales de objeto y matrices para almacenar datos. JSON representa solamente datos →No incluye el concepto de variables, asignaciones o igualdades. [33]

Este código JavaScript:

```

Var oPersona3 = [{"nombre":"Robert", "edad":30, "hijos": ["Jaime","Pepe","Alfonso"]},
{"nombre":"Maria", "edad":36, "hijos": ["Hijo Maria","Hijo2 Maria"]};
  
```

Quedaría en formato JSon como sigue:

```
[{"nombre":"Robert", "edad":30, "hijos": ["Jaime", "Pepe", "Alfonso"]}, {"nombre":"Maria", "edad":36, "hijos": ["Hijo Maria", "Hijo2 Maria"]}]
```

Se elimina la variable oPersona3, así como el punto y coma del final. Si se transmite esta información a través de HTTP a un navegador, será bastante rápido, dado el reducido número de caracteres.

#### 2.5.4 Algoritmo de Minería de Datos seleccionado

**Simple K-means:** Es el algoritmo de agrupamiento más conocido y se encuentra clasificado como un algoritmo particional. Está basado en la minimización de la distancia interna entre los elementos de los K grupos definidos por el usuario que realiza la minería. Este algoritmo permite el trabajo con atributos: numéricos, binarios, nominales, ordinales, funciona eficientemente con una gran cantidad de datos. [6]

##### Características del Simple K-means.

- Escalabilidad: normalmente el algoritmo corre con pocos datos.
- Manejo de ruido: sensible a datos erróneos.
- Grupos de formas arbitrarias: basado en distancias numéricas.
- Requerimientos mínimos para especificar parámetros, como el número de grupos.

##### Pasos para aplicar el algoritmo Simple K-means.

1. Se especifica la cantidad de grupos (K) que se van a crear y se selecciona de manera aleatoria los centros de cada grupo (k-centroides), a partir de la media que establece para cada uno de los elementos dentro de la vista minable.
2. Cada una de las instancias, es asignada al grupo con características similares más cercanas.
3. Se calcula las distancias de todos los puntos a los k-centroides y se forman k grupos, asignando a cada grupo el centroide más cercano.
4. Se repite el paso anterior, hasta que el valor de los centroides no varíe más, después de cada iteración.

Un buen método de agrupamiento va a producir grupos de alta calidad con alta similitud entre los elementos que se identifican en una misma clase y baja similitud entre clases diferentes. La calidad del resultado de un agrupamiento depende tanto de la medida de similitud utilizada por el método y su aplicación. El cálculo de las distancias de los elementos se utiliza normalmente para medir la similitud o disimilitud entre dos objetos de datos. [6]

Se seleccionó la distancia Euclidiana puesto que esta es efectiva y a su vez la más sencilla de calcular, mientras menos complejidad tenga el cálculo de la distancia más rápido se ejecuta el algoritmo.

En matemáticas, la distancia euclidiana o euclídea es la distancia "ordinaria" (que se mediría con una regla) entre dos puntos de un espacio euclídeo. Ejemplo en un espacio bidimensional, la distancia euclidiana entre dos puntos  $P_1$  y  $P_2$ , de coordenadas cartesianas  $(x_1, y_1)$  y  $(x_2, y_2)$  respectivamente, es:

Euclidiana

$$d_E(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

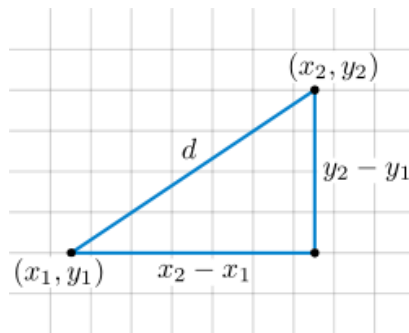


Figura 8. Distancia en un sistema de coordenadas cartesianas.

En general, la distancia euclidiana entre los puntos  $P = (p_1, p_2, \dots, p_n)$  y  $Q = (q_1, q_2, \dots, q_n)$ , del espacio euclídeo  $n$ -dimensional, se define como:

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Respecto a la complejidad computacional, el agrupamiento  $k$ -means para problemas en espacios de  $d$  dimensiones es:  **$O(n^{dk+1} \log n)$** , puesto que  $k$  y  $d$  siempre van a ser definidos en la solución,

donde  $n$  es el número de entidades a particionar,  $k$  es el número de grupos y  $d$  toma el valor de cada rejilla en una entidad de la vista minable. [35]

### **2.5.5 Herramienta seleccionada para la Minería de Datos**

WEKA es un programa de Minería de Datos, libre y de código abierto. Se utiliza en la construcción de esta solución para garantizar un mayor rendimiento, debido a que cuenta con las siguientes características:

- Amplia implementación de algoritmos, incluso con variaciones de los mismos para problemas específicos.
- Se integra con varias plataformas de desarrollo.
- Permite que el usuario haga modificaciones en el código fuente, brindando, no solo, facilidad para usar las herramientas de GUI, sino que también permite ejecutar simulaciones de algoritmos estándares.

### **2.5.6 Lenguaje de programación**

#### **Java.**

Es un lenguaje de programación multiplataforma, es decir, es ejecutable en la mayoría de los sistemas operativos, incluso en sistemas operativos de móviles. Hereda la sintaxis de C/C++ y muchas de las características orientadas a objetos de C++. Es un lenguaje de distribución libre, simple, orientado a objetos, robusto, de arquitectura neutral, seguro, portable, interpretado, multihilo y dinámico.

Como lenguaje de programación java es un lenguaje de alto nivel, que permite el desarrollo tanto de arquitecturas cliente-servidor como de aplicaciones distribuidas, consistentes en crear aplicaciones capaces de conectarse a otros computadores y ejecutar tareas en varios computadores simultáneamente, repartiendo por lo tanto el trabajo. Aunque otros lenguajes de programación también permiten crear aplicaciones de este tipo, Java incorpora en su propio API estas funcionalidades. [17]

#### **C++.**

Es un lenguaje imperativo orientado a objetos derivado del C. En realidad es un súper conjunto de C, que nació para añadirle cualidades y características de las que carecía. El resultado es que como su ancestro, sigue muy ligado al hardware subyacente, manteniendo una considerable

potencia para programación a bajo nivel, pero se la han añadido elementos que le permiten también un estilo de programación con alto nivel de abstracción [18].

Es un lenguaje complicado debido a que no cuenta con un respaldo de Frameworks que contribuyan a minimizar el tiempo de desarrollo de las funcionalidades. Además la reutilización de código en forma de librería de usuario es otra de sus propiedades. El código es transportable, es decir, un programa en ANSI en C++ podrá ejecutarse en cualquier máquina y bajo cualquier sistema operativo [19]. Pero este lenguaje de programación presenta algunas desventajas, sobre todo para el desarrollo orientado a Internet, donde la implementación de sistemas con esta característica se torna difícil y de baja productividad. Además el lenguaje cuenta con una curva de aprendizaje compleja.

### **Python.**

Es un lenguaje de programación similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de script, con tipado dinámico, multiplataforma y orientado a objetos. Python tiene muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. También permite la programación imperativa, programación funcional y programación orientada a aspectos. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, hacen que desarrollar una aplicación en Python sea sencillo y muy rápido. Su sintaxis es tan sencilla y cercana al lenguaje natural, que los programas elaborados en Python parecen pseudocódigo. Por este motivo se trata además de uno de los mejores lenguajes para comenzar a programar. Sin embargo no es recomendado para la programación de bajo nivel o para aplicaciones en las que el rendimiento sea crítico. [21]

Luego de un estudio detallado de cada uno de los lenguajes de programación, se selecciona el lenguaje Java. Debido a la alta productividad comprobada en los desarrollados, sobre todo para aplicaciones altamente interactivas bajo la modalidad de Web. Permite la reutilización y ejecución de los algoritmos de WEKA mediante una integración directa con su biblioteca. Con este lenguaje se logran aplicaciones independientes de la plataforma. Además permite un fácil desarrollo tanto de arquitecturas cliente-servidor como de aplicaciones distribuidas. Se crean soluciones modulares y códigos reutilizables.

## 2.5.7 Entorno de desarrollo

**Netbeans:** Es un entorno de desarrollo gratuito y de código abierto que permite el uso de un amplio rango de tecnologías de desarrollo tanto para escritorio, como aplicaciones Web. Da soporte a las siguientes tecnologías: **Java, PHP, Groovy, C/C++, HTML5**, entre otras. Además puede instalarse en varios sistemas operativos: Windows, Linux, Mac OS, etc. Suele dar soporte a casi todas las novedades en el lenguaje **Java**. Cualquier vista previa del lenguaje es rápidamente soportada por Netbeans. Tiene un excelente balance entre una interfaz con múltiples opciones y un aceptable completamiento de código. [22]

### Características de Netbeans [22]

- Buen **editor de código, multilenguaje**, con sugerencias de código, acceso a clases pinchando en el código, control de versiones, localización de ubicación de la clase actual, comprobaciones sintácticas y semánticas, etcétera.
- Simplifica la **gestión de grandes proyectos** con el uso de diferentes vistas, asistentes de ayuda, y estructurando la visualización de manera ordenada, lo que ayuda en la productividad.
- Herramientas para **depurado de errores**: el depurador que incluye el IDE es bastante útil para encontrar dónde fallan las cosas.
- **Optimización de código**: por su parte el **Profiler** ayuda a optimizar las aplicaciones e intenta hacer que se ejecuten más rápido y con el mínimo uso de memoria.
- **Acceso a base de datos**: el Netbeans permite la conexión a distintos sistemas gestores de bases de datos, como pueden ser Oracle, PostgreSQL y demás, se pueden ver las tablas, realizar consultas y modificaciones, y todo ello integrado en el propio IDE.
- Se integra con diversos **servidores de aplicaciones**, de tal manera que se puede gestionar desde el propio IDE: inicio, parada, arranque en modo debug, despliegues. Se puede usar Apache Tomcat, GlassFish, JBoss, WebLogic, Sailfin, Sun Java System Application Server.



## 2.5.8 Metodología de Desarrollo de Software

Una metodología de desarrollo de software se refiere a una forma de trabajo que es usada para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. Es un conjunto de pasos y procedimientos a seguir con el objetivo de aumentar la calidad del software creado.

A pesar de la variedad de metodologías usadas, se ha comprobado que muy pocos proyectos de la universidad la aplican en su totalidad. Las diferencias entre estas metodologías no radica únicamente en los productos de trabajos que proponen o en sus roles, sino en su forma de planificar el proyecto y realizar las estimaciones del tiempo. Factor determinante en la culminación exitosa de todo desarrollo de software, por lo que uno de los principales problemas detectados es que sin importar la metodología que se usa se está planificando con un único cronograma tipo, además de forzar el método de estimación definido en la universidad y que responde en su gran mayoría a la metodología RUP. [24]

Para lograr erradicar los problemas detectados, se decide escoger una metodología para ser adaptada a lo que ya la Universidad ha estado proponiendo como ciclo de vida de los proyectos, sin alejarse de lo que hasta el momento se ha trabajado e introducir la menor cantidad de cambios posibles. [24]

El Proceso Unificado Ágil es una versión simplificada del Proceso Unificado de Racional (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. [24]

El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (Test Driven Development - TDD en inglés).
- Modelado ágil.
- Gestión de Cambios ágil.
- Refactorización de base de datos para mejorar la productividad.

Fases de AUP: [24]

- **Inicio:** En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

- **Ejecución:** En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente.
- **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

## 2.5.9 Herramienta CASE

Como herramienta para modelar los diagramas planteados durante el desarrollo del trabajo se utilizó Visual Paradigm for UML, una herramienta CASE que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. [27]

### 2.5.10 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptado para resolver un problema de diseño general en un contexto particular. En la construcción de la solución, para garantizar una buena práctica de programación se utilizaron los siguientes patrones:

Patrones de Software para la Asignación General de Responsabilidad (GRASP, por sus siglas en inglés). Estos patrones describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades [37]. Dentro de los patrones de diseño GRASP se hizo uso de los patrones:

**Experto:** El propósito del patrón Experto es asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad.

El patrón experto se percibe en las clases SmartServer, Manager y Analysis debido a que la clase SmartServer delega la responsabilidad de gestionar las tareas a la clase Manager y la misma a su vez le asigna el análisis a la clase Analysis.

**Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que debe conectar con el objeto producido en cualquier evento. Al escoger este patrón, se da soporte al bajo acoplamiento.

El patrón creador se evidencia en las clases que representan una entidad que tiene la responsabilidad de crear otra, tal es el caso de la clase Manager que se encarga de crear instancias de las clases Notification, Task y Analysis, debido a que estas clases contienen la información para crear notificaciones, tareas y para ejecutar el algoritmo de análisis.

Los patrones pandilla de los cuatro (GOF, por sus siglas en inglés) o comúnmente llamados Patrones GOF, se caracterizan por ser soluciones concretas, se utilizan en situaciones frecuentes y favorecen la reutilización de código [39]. Dentro del grupo de los patrones GOF se utilizó el patrón Singleton.

Singleton: El patrón de diseño Singleton (instancia única) para gestionar múltiples sesiones, está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella. Este patrón se implementa creando en una clase un método que realiza una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado)

El uso de este patrón se puede observar en la clase SmartServer, esta clase crea una instancia para la gestión de múltiples sesiones.

Los patrones de Alta cohesión fueron de ayuda para definir las clases con responsabilidades estrechamente relacionadas de modo que no realicen un trabajo enorme.

## **2.7 Diagrama de Clases**

El diagrama de clases es utilizado para describir la estructura del sistema mostrando sus clases y la relación entre estas.

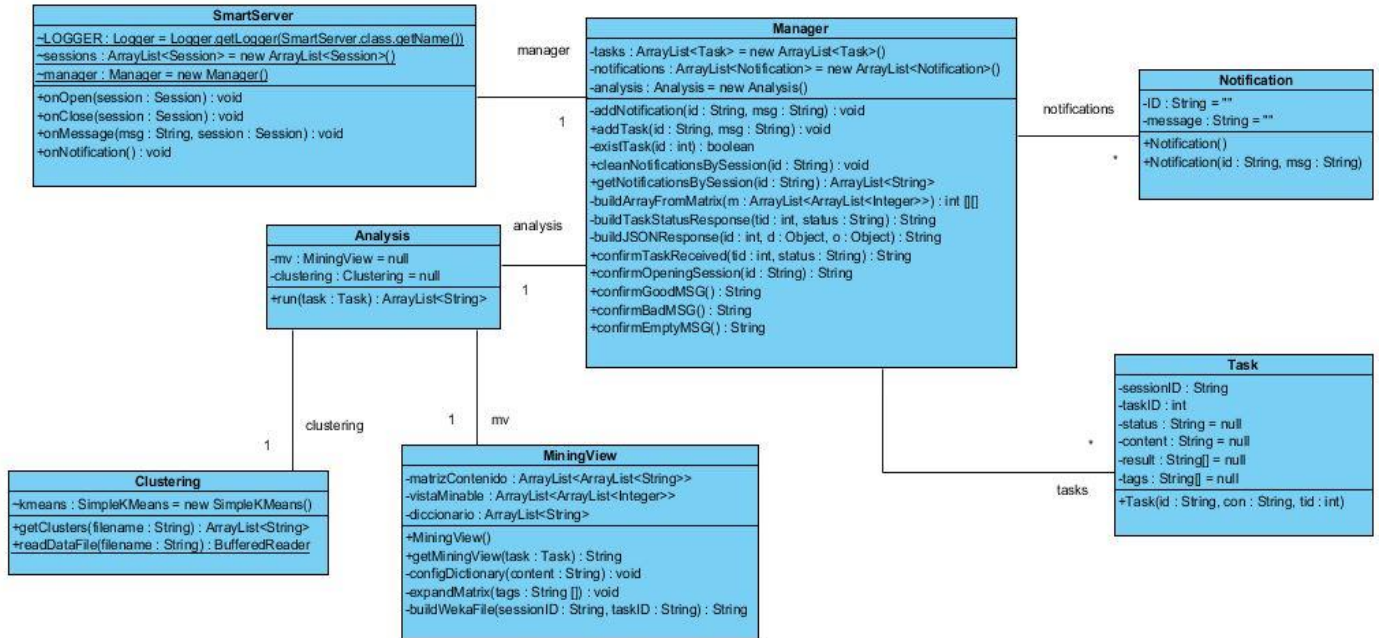


Figura 9. Diagrama de Clases.

## 2.8 Arquitectura de la solución

Como patrón arquitectónico se define la arquitectura 4 capas. Para modelar su distribución se empleó el diagrama de paquetes debido a que estos muestran cómo un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema.

Se utilizó la arquitectura de 4 capas por la forma en que se separan los diferentes aspectos del desarrollo, lo que permite que las distintas partes de la aplicación se puedan modificar sin que se afecten las demás capas implementadas en el desarrollo. La cual describe la separación de las funcionalidades de la misma manera que el estilo de capas. Las capas se caracterizan por la descomposición funcional de los componentes y/o servicio de la aplicación, lo que provee escalabilidad, disponibilidad, manejabilidad, mantenibilidad y reutilización. Cada capa es independiente de las demás.

En el sistema implementado se tiene físicamente 4 paquetes donde el paquete server contiene la clase SmartServer que maneja las funcionalidades puntuales de comunicación entre el servidor y el cliente. La clase Manager está contenida en el paquete controller. Mientras que en entities están contenidas las clases Task y Notification que definen los objetos correspondientes a las mismas. Por último se encuentran agrupadas las clases Analysis, Clustering y MiningView que intervienen en el análisis de los datos en el paquete Auxiliar.

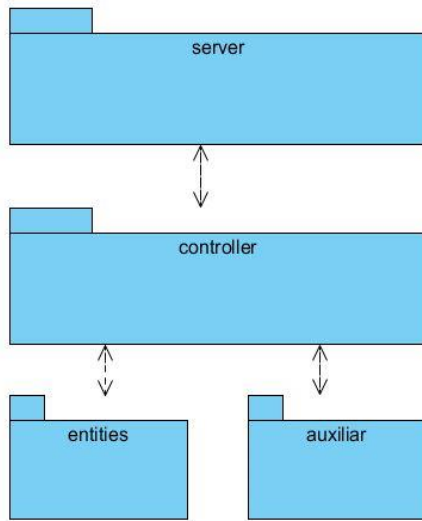


Figura 10. Arquitectura 4-capas.

## 2.9 Diagrama de Colaboración

El Diagrama de Colaboración muestra el comportamiento de los objetos identificados en el modelo de clase y ofrece una mejor visión espacial, mostrando los enlaces de comunicación entre estos objetos para comprender la participación de un objeto en el sistema.

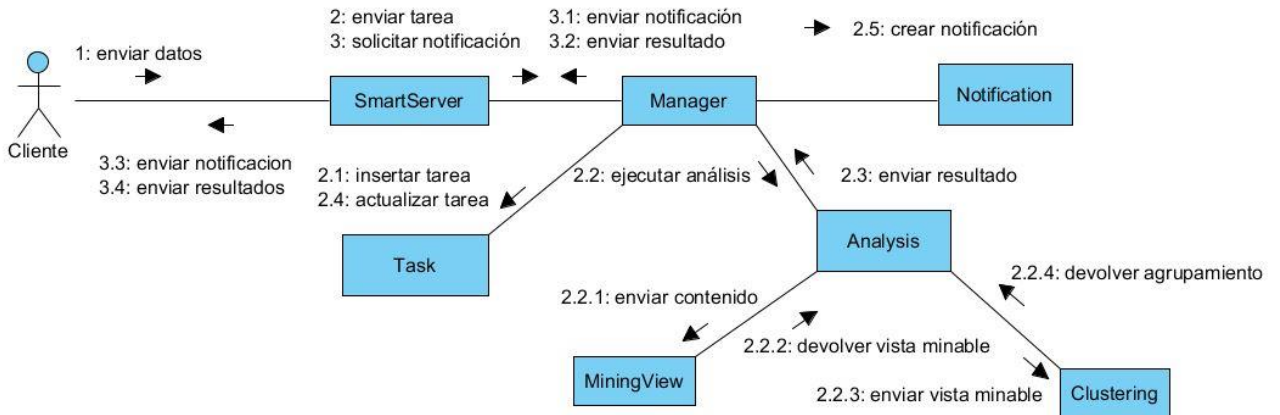


Figura 11. Diagrama de Colaboración.

## **2.10 Conclusiones del Capítulo**

En este capítulo se realizó el diseño de la solución empleando el modelado ágil de la metodología AUP, mediante las tarjetas CRC, las historias de usuarios, el diagrama de clases y el diagrama de colaboración obteniéndose los artefactos que evidencia el desarrollo de la solución.

Se definieron los requisitos y la arquitectura para dar cumplimiento a la fase de ejecución de la metodología. Se explicó de forma breve el proceso de negocio y por último se modeló el dominio en relación con la arquitectura de solución propuesta.

## CAPÍTULO 3: CONSTRUCCIÓN Y VALIDACIÓN

### Introducción

En este capítulo se seleccionan las tecnologías adecuadas para dar cumplimiento a la solución propuesta. Se validan las funcionalidades y se hacen las pruebas pertinentes para comprobar los resultados obtenidos. Luego de un análisis detallado de los lenguajes de programación se realiza una selección del mismo. Además se propone una descripción del algoritmo de Minería de Datos empleado para dar solución al problema y se define la metodología de desarrollo.

### 3.1 Diagrama de Componentes

Los diagramas de componentes muestran como el sistema se encuentra dividido en componentes y la relación que existe entre ellos. Además, ayudan a los desarrolladores a visualizar el camino de la implementación. Teniendo en cuenta la arquitectura definida y el modelo de desarrollo a utilizar se definen como estructura básica los componentes.

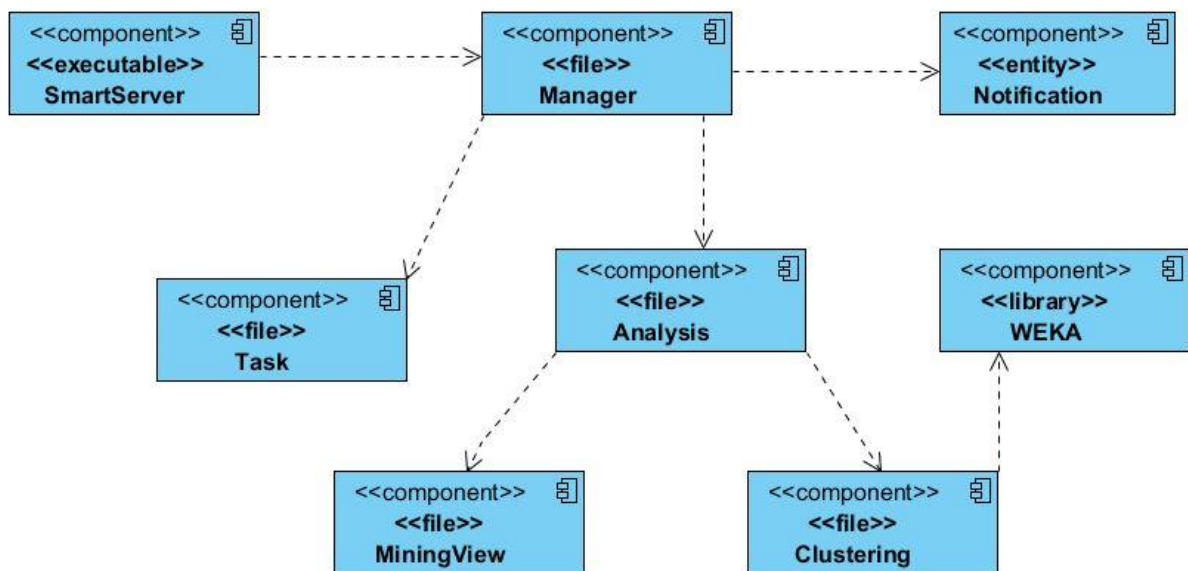


Figura 12. Diagrama de Componentes.

De los estereotipos que define UML se aplican en los componentes:

- Executable, especifica un componente que se puede ejecutar en un nodo.
- Library, especifica una biblioteca de objetos estática o dinámica.
- File, especifica un componente que representa un documento que contiene código fuente o datos.

- Entity, especifica una clase que cuenta con las propiedades necesarias para la definición de la instancia de sus objetos.

### 3.1.2 Diagrama de Despliegue

El diagrama de despliegue es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes. Un diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema final, es decir, la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software (procesos y objetos que se ejecutan en ellos).

Debido a que se quiere realizar una solución centralizada diseñada para el servidor de análisis de tarea, la solución se basa en el despliegue de un sistema que emplee comunicación a través del protocolo de tcp/ip entre el cliente y el servidor.

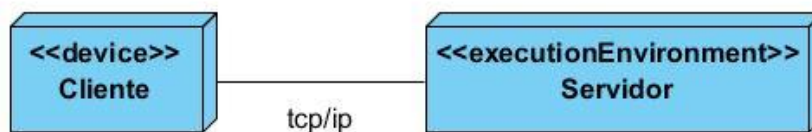


Figura 13. Diagrama de despliegue.

### 3.2 Estándar de código

Con el objetivo de facilitar el mantenimiento de una aplicación, mejorar la legibilidad del código y al mismo tiempo permitir su compresión rápida por cualquier programador se establecen los estándares recomendados por Sun Microsystems empleados en el desarrollo de software sobre la plataforma Java. [26]

#### Organización de ficheros

- Las clases en Java se agrupan en paquetes.
- Estos paquetes se deben organizar de manera jerárquica.
- Dentro del paquete principal las clases se organizarán en subpaquetes en función del área, organismo o sección al que pertenezca el código desarrollado.



- Un fichero consta de secciones que deben estar separadas por líneas en blanco y comentarios opcionales que identifiquen cada sección.
- Deben evitarse los ficheros de gran tamaño que contengan más de 1000 líneas.

### **Fichero fuente Java (.java)**

Cada fichero fuente Java debe contener una única clase o interfaz pública. El nombre del fichero tiene que coincidir con el nombre de la clase. Cuando existan varias clases privadas asociadas funcionalmente a una clase pública, podrán colocarse en el mismo fichero fuente que la clase pública. La clase pública debe estar situada en primer lugar dentro del fichero fuente.

En todo fichero fuente Java se distinguen las siguientes secciones: comentarios de inicio, sentencia de paquete, sentencias de importación y declaraciones de clases e interfaces.

### **Comentarios de inicio**

Todo fichero fuente debe comenzar con un comentario que incluya el nombre de la clase, información sobre la versión del código, la fecha y el copyright. El copyright indica la propiedad legal del código, el ámbito de distribución, el uso para el que fue desarrollado y su modificación.

### **Sentencias de paquete**

La primera línea no comentada de un fichero fuente debe ser la sentencia de paquete, que indica el paquete al que pertenece(n) la(s) clase(s) incluida(s) en el fichero fuente. Tras la declaración del paquete se incluirán las sentencias de importación de los paquetes necesarios.

### **Declaración de clases / interfaces**

Durante el desarrollo de clases / interfaces se deben seguir las siguientes reglas:

- No incluir ningún espacio entre el nombre del método y el paréntesis inicial del listado de parámetros.
- El carácter inicio de bloque ("{"") debe aparecer al final de la línea que contiene la sentencia de declaración.
- El carácter fin de bloque ("}") se sitúa en una nueva línea tabulada al mismo nivel que su correspondiente sentencia de inicio de bloque, excepto cuando la sentencia sea nula, en tal caso se situará detrás de "{".
- Los métodos se separarán entre sí mediante una línea en blanco.

## **Inicialización**

Toda variable local tendrá que ser inicializada en el momento de su declaración, salvo que su valor inicial dependa de algún valor que tenga que ser calculado previamente.

## **Sentencias**

Cada línea debe contener como máximo una sentencia. Las sentencias pertenecientes a un bloque de código estarán tabuladas un nivel más a la derecha con respecto a la sentencia que las contiene.

La sentencia "try/catch" siempre debe tener el formato siguiente,

```
try {  
    sentencias;  
} catch (ClaseException e) {  
    sentencias;  
}
```

## **Visibilidad de atributos de instancia y de clase**

Los atributos de instancia y de clase serán siempre privados, excepto cuando tengan que ser visibles en subclases herederas, en tales casos serán declarados como protegidos. El acceso a los atributos de una clase se realizará por medio de los métodos "get" y "set" correspondientes.

## **3.3 Validación de funcionalidades**

La validación es el proceso de evaluar cuál sería el rendimiento de los modelos de Minería de Datos con datos reales. Es importante que se validen los modelos de Minería de Datos entendiendo su calidad y sus características antes de implementarlos en un entorno de producción.

La correcta realización de las técnicas empleadas depende de la estandarización de los datos. Estos al ser extraídos se caracterizan por ser una serie de cadenas de caracteres. Por lo cual se decide realizar una nominalización de los mismos para ajustarlos a formatos definidos para análisis estadísticos de la herramienta empleada para la Minería de Datos WEKA.

Para realizar análisis en WEKA el archivo generado debe tener la siguiente estructura:

```

1  %
2  @relation descripcion
3  @attribute a {'0','1'}
4  @attribute b {'0','2'}
5  @attribute c {'0','3'}
6  @attribute d {'0','4'}
7  @attribute e {'0','5'}
8  @attribute f {'0','6'}
9  @attribute g {'0','7'}
10 @attribute h {'0','8'}
11 @attribute i {'0','9'}
12 @data
13 '1','2','0','0','5','6','0','8','0'
14 '0','2','0','0','0','6','7','0','9'
15 '1','2','0','4','0','6','0','0','0'
16 '1','2','0','0','0','0','7','0','9'
17 '0','0','3','0','0','0','7','8','9'
18 '1','2','3','4','5','0','0','0','9'
19 '1','2','3','4','0','6','0','0','9'
20 '1','2','3','0','0','6','0','8','0'
21 '1','2','3','0','0','6','7','8','0'
22 '0','2','0','0','5','0','7','0','0'
23 '1','0','0','0','5','0','7','0','9'
24 '1','2','0','4','0','6','7','8','9'
25 '1','2','3','4','5','6','0','8','9'
26 '1','2','3','4','5','6','0','0','0'

```

Figura 14. Archivo con formato arff para el uso de la herramienta WEKA.

Una vez que los datos son cargados en la herramienta se aplica el algoritmo de agrupamiento K-means con una distancia Euclidiana. Luego de aplicar el algoritmo K-means previamente configurado para que identifique un máximo de grupos 10, el resultado obtenido muestra ocho grupos, como se puede apreciar en la siguiente figura:

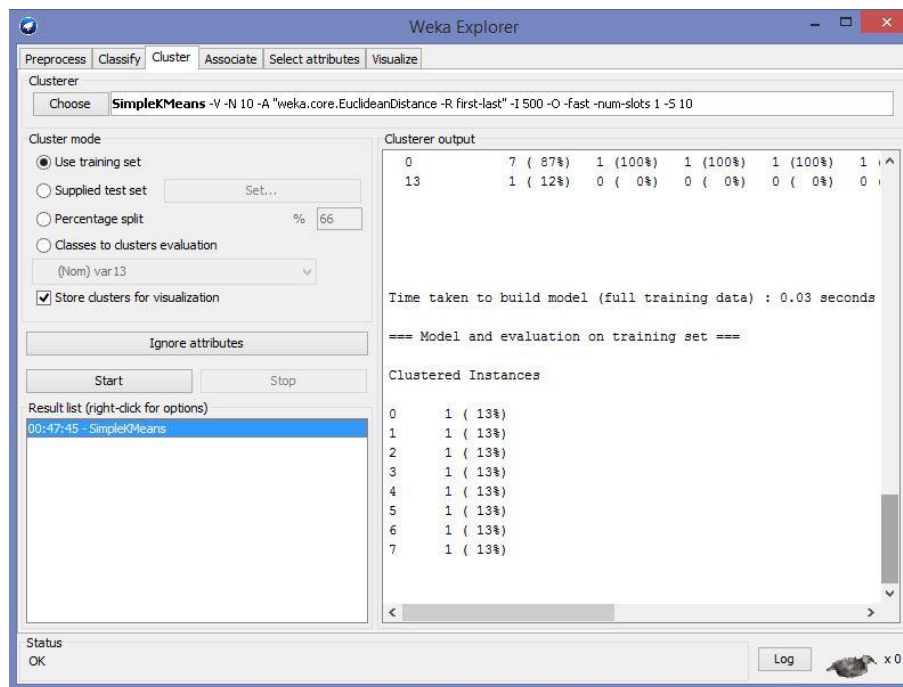


Figura 15. Resultado del Algoritmo K-means en la herramienta WEKA utilizando la distancia Euclidiana.

Tras la ejecución del algoritmo K-means, WEKA visualiza el resultado de la siguiente manera:

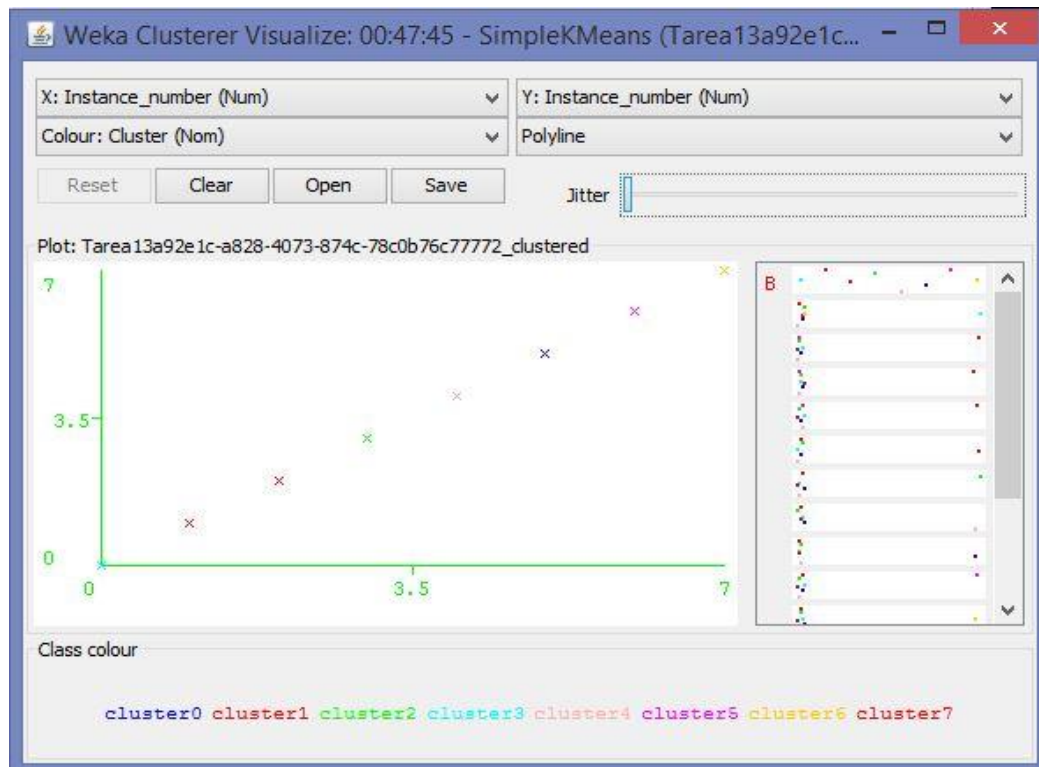


Figura 16. Visualización de la asignación del clúster utilizando la distancia Euclidiana.

### 3.4 Pruebas y resultados obtenidos

La etapa de implementación dentro del proceso de desarrollo del software se dividió en tres iteraciones. En la primera iteración se implementaron las funcionalidades encargadas de la gestión de las sesiones y de las tareas. En la segunda iteración se desarrollaron las funcionalidades que gestionan las notificaciones y en una tercera iteración se hicieron las funcionalidades encargadas del agrupamiento de los datos. A cada una de estas iteraciones se le realizaron pruebas con el fin de conocer el funcionamiento interno del producto.

#### 3.4.1 Plan de pruebas y Diseños de casos de pruebas

##### Plan de prueba para la Historia de Usuario 1: Adicionar sesión

Descripción General: Se encarga de adicionar una sesión al gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe seleccionar la opción Conectar al Servidor en la parte superior izquierda de la interfaz principal BISA Smart Client.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU1: Adicionar sesión	Escenario 1.1: Inicio de sesión satisfactorio.	Se crea la sesión en el servidor.	1 Seleccionar la opción Conectar al Servidor en la parte superior izquierda de la interfaz principal BISA Smart Client. 2 En la ventana LOGS DESDE EL SERVER aparece el identificador que le asigna el servidor al cliente y un mensaje que indica que la sesión se ha creado Session Opened.
	Escenario 1.2: Inicio de sesión insatisfactorio.	El servidor no se encuentra disponible.	1 Seleccionar la opción Conectar al Servidor en la parte superior izquierda de la interfaz principal BISA Smart Client. 2 En la ventana LOGS DESDE EL SERVER aparece una notificación de error de conexión.

Tabla 12: Plan de prueba para la Historia de Usuario 1. Adicionar sesión.

### Diseño de caso de prueba 1: Adicionar sesión

A: Botón accionado, NA: Botón no accionado.

Id de la sección	Escenario	Variable 1 Botón Conectar al servidor	Respuesta del sistema	Resultado de la prueba
P.HU1	Adicionar sesión	A	Se adiciona la sesión	Prueba satisfactoria
		NA	No se adiciona la sesión	
		A	No se adiciona la sesión	Prueba no satisfactoria

Tabla 14: Diseño de caso de prueba para la Historia de Usuario 1. Adicionar sesión.

## Plan de prueba para la Historia de Usuario 2: Eliminar sesión

Descripción General: Se encarga de eliminar una sesión en el gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe seleccionar la opción Desconectar en la parte superior izquierda de la interfaz principal BISA Smart Client.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU2: Eliminar sesión	Escenario 2.1: Cierre de sesión satisfactorio.	Se elimina la sesión en el servidor.	1 Seleccionar la opción Desconectar en la parte superior izquierda de la interfaz principal BISA Smart Client. 2 En la ventana LOGS DESDE EL SERVER aparece la notificación Session Closed indicando que la sesión fue cerrada.

Tabla 13: Plan de prueba para la Historia de Usuario 2. Eliminar sesión.

## Diseño de caso de prueba 2: Eliminar sesión

A: Botón accionado, NA: Botón no accionado.

Id de la sección	Escenario	Variable 1 Botón Desconectar	Respuesta del sistema	Resultado de la prueba
P.HU2	Eliminar sesión	A	Se elimina la sesión	Prueba satisfactoria
		NA	No se elimina la sesión.	

Tabla 14: Diseño de caso de prueba para la Historia de Usuario 2. Eliminar sesión.

## Plan de prueba para la Historia de Usuario 3: Insertar Tarea

Descripción General: Se encarga de insertar una tarea en el gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe seleccionar la opción Agregar Tarea en la parte superior izquierda de la interfaz principal BISA Smart Client.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU3: Insertar Tarea	Escenario 3.1: Inserción de tarea satisfactoria.	Se inserta el nombre de la tarea, las diferentes etiquetas y el contenido para analizar y se envía al servidor de análisis.	<p>1 Seleccionar la opción Agregar Tarea en la parte superior izquierda de la interfaz principal BISA Smart Client.</p> <p>2 En la ventana EDITAR DATOS DE LA TAREA insertar nombre de la tarea, las diferentes etiquetas y el contenido para analizar.</p> <p>3 Seleccionar la opción Enviar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.</p> <p>4 En la ventana LOGS DESDE EL SERVER una vez que la tarea se ha insertado aparece la notificación Pendiente.</p>
	Escenario 3.2: Datos en blanco.	En caso de no insertar el contenido.	<p>1 Seleccionar la opción Agregar Tarea en la parte superior izquierda de la interfaz principal BISA Smart Client.</p> <p>2 En la ventana EDITAR DATOS DE LA TAREA insertar nombre de la tarea y las diferentes etiquetas.</p> <p>3 Seleccionar la opción Enviar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.</p> <p>4 En la ventana LOGS DESDE EL SERVER aparece la notificación BAD indicando que la tarea no se ha insertado por algún error.</p>

Tabla 15: Plan de prueba para la Historia de Usuario 3. Insertar Tarea.

### Diseño de caso de prueba 3: Insertar Tarea.

C: Datos correctos, V: Campo vacío.

Id de la sección	Esce nario	Variable 1 Nombre de Tarea	Variable 2 Contenido	Variable 3 Etiquetas	Respues ta del sistema	Resultado de la prueba
P.HU3	Insertar Tarea	c	c	c	Se inserta la tarea	Prueba satisfactoria
		v	c	c	Se inserta la tarea	
		c	c	v	Se inserta la tarea	
		c	v	c	No se inserta la tarea	
		c	v	c	Se inserta la tarea	Prueba no satisfactoria

Tabla 16: Diseño de caso de prueba para la Historia de Usuario 3. Insertar Tarea.

### Plan de prueba para la Historia de Usuario 5: Eliminar Tarea

Descripción General: Se encarga de Eliminar una tarea en el gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe seleccionar la opción Eliminar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU5: Eliminar Tarea	Escenario 5.1: Eliminación de tarea satisfactoria.	Se elimina la tarea en el servidor de análisis.	1 Seleccionar la opción Eliminar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client. 2 Se elimina la tarea.

Tabla 17: Plan de prueba para la Historia de Usuario 5. Eliminar Tarea.

### Diseño de caso de prueba 5: Eliminar Tarea.

A: botón accionado NA: botón no accionado

Id de la sección	Escenario	Variable 1	Respuesta del	Resultado de la
------------------	-----------	------------	---------------	-----------------



		Botón Eliminar Tarea.	sistema	prueba
P.HU5	Eliminar tarea	A	Se elimina la tarea	Prueba satisfactoria
		NA	No se elimina la tarea	
		A	No se elimina la tarea	Prueba no satisfactoria

Tabla 18: Diseño de caso de prueba para la Historia de Usuario 5. Eliminar Tarea.

### Plan de prueba para la Historia de Usuario 6: Crear notificaciones

Descripción General: Se encarga de Crear notificaciones en el gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe seleccionar las siguientes opciones: Conectar al servidor, Desconectar o Enviar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU6: Crear notificaciones	Escenario 6.1: Crea notificaciones satisfactoriamente.	Se crean notificaciones en el servidor de análisis.	1 Seleccionar la opción Conectar al servidor, Desconectar o Enviar Tarea en la interfaz principal BISA Smart Client. 2 En el servidor de análisis se crean las notificaciones correspondientes a las acciones seleccionadas por el cliente.
	Escenario 6.2: No se crean notificaciones	En caso de eliminar una tarea.	1 Seleccionar la opción Eliminar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.

Tabla 19: Plan de prueba para la Historia de Usuario 6. Crear Notificaciones.

### Diseño de caso de prueba 6: Crear Notificaciones.

A: botón accionado NA: botón no accionado

Id de la sección	Escenario	Variable 1 Botón Conectar al	Variable 2 Botón Desconectar	Variable 3 Eliminar tarea	Respuesta del sistema	Resultado de la prueba
------------------	-----------	------------------------------------	------------------------------------	---------------------------------	--------------------------	---------------------------

		servidor				
P.HU6	Crear Notificaciones	A	A	NA	Se crea la notificación	Prueba satisfactoria
		NA	NA	A	No se crea la notificación	
		NA	NA	A	Se crea la notificación	Prueba no satisfactoria

Tabla 20: Diseño de caso de prueba para la Historia de Usuario 6. Crear Notificaciones.

### Plan de prueba para la Historia de Usuario 8: Eliminar notificaciones

Descripción General: Se encarga de Eliminar notificaciones en el gestor de tareas de análisis.

Condiciones de Ejecución: El servidor debe eliminar las notificaciones una vez que las mismas son enviadas al cliente.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU8: Eliminar notificaciones	Escenario 8.1: Eliminar notificaciones satisfactoriamente.	Se eliminan las notificaciones enviadas en el servidor de análisis.	1 El servidor elimina las notificaciones una vez que son enviadas a los clientes.

Tabla 21: Plan de prueba para la Historia de Usuario 8. Eliminar Notificaciones.

### Diseño de caso de prueba 8: Eliminar Notificaciones.

Cant: cantidad de elementos en la lista de notificación.

Id de la sección	Escenario	Variable 1 LOGS EN EL SERVER	Respuesta del sistema	Resultado de la prueba
P.HU8	Eliminar Notificaciones	Cant	Se elimina la notificación	En el servidor se muestra el valor de la variable Cant con un valor menor al que tenía anteriormente.
		Cant	No se elimina la notificación	En el servidor se muestra el valor de la variable Cant con un valor igual o mayor al que tenía anteriormente.

Tabla 22: Diseño de caso de prueba para la Historia de Usuario 8. Eliminar Notificaciones.

### Plan de prueba para la Historia de Usuario 9: Crear diccionario

Descripción General: Se encarga de crear un diccionario de datos en el gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe enviar los datos a los cuales les desea realizar análisis.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU9: Crear diccionario	Escenario 9.1: Se crea el diccionario satisfactoriamente.	Se crea un diccionario en el servidor.	1 El servidor utiliza el contenido de la tarea recibida para conformar el diccionario de la vista minable.

Tabla 23: Plan de prueba para la Historia de Usuario 9. Crear diccionario.

### Diseño de caso de prueba 9: Crear diccionario.

C: Datos correctos, V: Campo vacío.

Id de la sección	Escenario	Variable 1 Nombre de Tarea	Variable 2 Contenido	Variable 3 Etiquetas	Respuesta del sistema	Resultado de la prueba
P.HU9	Crear diccionario.	c	c	c	Se crea el diccionario.	Prueba satisfactoria
		v	c	c	Se crea el diccionario.	
		c	c	v	Se crea el diccionario.	
		c	c	c	No se crea el diccionario.	Prueba no satisfactoria

Tabla 25: Diseño de caso de prueba para la Historia de Usuario 9. Crear diccionario.

### Plan de prueba para la Historia de Usuario 11: Ejecutar algoritmo de análisis

Descripción General: Se encarga de Ejecutar el algoritmo de análisis en el gestor de tareas de análisis.

Condiciones de Ejecución: En el servidor debe existir al menos una vista minable correspondiente a la tarea que se analiza.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU11: Ejecutar algoritmo de análisis	Escenario 11.1: Ejecuta el algoritmo de análisis	Se ejecuta el algoritmo de análisis en el servidor, luego de conformar la vista minable.	1 El servidor le envía la vista minable al algoritmo de análisis. 2 Ejecuta el algoritmo. 3 Se obtienen los grupos resultantes.

Tabla 24: Plan de prueba para la Historia de Usuario 11. Ejecutar algoritmo de análisis.

### Diseño de caso de prueba 11: Ejecutar algoritmo de análisis.

Id de la sección	Escenario	Respuesta del sistema	Resultado de la prueba
P.HU11	Ejecutar algoritmo de análisis	Se ejecuta el algoritmo de análisis	El servidor muestra en la consola los grupos obtenidos.

Tabla 26: Diseño de caso de prueba para la Historia de Usuario 11. Ejecutar algoritmo de análisis.

### Plan de prueba para la Historia de Usuario 12: Etiquetar grupos

Descripción General: Se encarga de Etiquetar los grupos generados por el algoritmo de análisis en el gestor de tareas de análisis.

Condiciones de Ejecución: El algoritmo de análisis debe generar los grupos correspondientes a los datos contenidos en la vista minable.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU12: Etiquetar grupos	Escenario 12.1: Etiquetar grupos	Se etiquetan los grupos generados por el algoritmo de análisis	1El servidor establece una relación entre los grupos obtenidos y los elementos analizados. 2 Establece una etiqueta para cada uno de los grupos.

Tabla 25: Plan de prueba para la Historia de Usuario 12. Etiquetar grupos.

## Diseño de caso de prueba 12: Etiquetar grupos

C: Datos correctos, V: Campo vacío.

Id de la sección	Escenario	Respuesta del sistema	Resultado de la prueba
P.HU12	Etiquetar grupos	Se etiquetan los grupos	El servidor muestra en la consola los grupos etiquetados.

Tabla 26: Diseño de caso de prueba para la Historia de Usuario 12. Etiquetar grupos.

## Plan de prueba para la Historia de Usuario 13: Enviar grupos

Descripción General: Se encarga de Enviar los grupos generados **por** el algoritmo de análisis en el gestor de tareas de análisis.

Condiciones de Ejecución: Los grupos generados por el algoritmo de análisis deben de estar etiquetados.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU13: Enviar grupos	Escenario 13.1: Enviar grupos	Se envían los grupos al cliente	1 El servidor establece un mensaje con los grupos etiquetados y los añade a la lista de notificaciones. 2 El cliente recibe el resultado mediante una notificación.

Tabla 27: Plan de prueba para la Historia de Usuario 13. Enviar grupos.

## Diseño de caso de prueba 13: Enviar grupos

C: Datos correctos, V: Campo vacío.

Id de la sección	Escenario	Variable 1 Resultados del agrupamiento	Respuesta del sistema	Resultado de la prueba
P.HU13	Enviar grupos	C	Se envían los grupos	Satisfactoria

Tabla 28: Diseño de caso de prueba para la Historia de Usuario 13. Enviar grupos.

### Desarrollo de las iteraciones de pruebas

Se definieron planes de pruebas y se diseñaron casos de prueba para cada una de las historias de usuario para asegurar que cada funcionalidad cumpliera con las exigencias del cliente. En cada iteración se realizaron pruebas a todas las funcionalidades en busca de no conformidades.

Durante las pruebas en la primera iteración se encontraron 4 no conformidades, las cuales fueron resueltas inmediatamente.

Durante las pruebas en la segunda iteración se encontraron 3 no conformidades, las cuales fueron resueltas inmediatamente.

Durante las pruebas en la tercera iteración se encontraron 6 no conformidades.

### 3.5 Pruebas de Aceptación

Las pruebas de aceptación son conocidas como pruebas de caja negra, tienen como fin validar que el sistema cumple con los requisitos básicos de funcionamiento esperado y permitir que el referente de negocio determine su aceptación. Las pruebas de aceptación aplicadas a la herramienta de análisis de datos fueron todas satisfactorias debido a que el desarrollo del producto se dividió en tres iteraciones con el fin de identificar y corregir las no conformidades.

Caso de Prueba de Aceptación	
Código: H1-P1	Historia de Usuario: Adicionar sesión.
Nombre: Adicionar sesión.	
Descripción: Prueba para la funcionalidad que permite guardar una nueva conexión en la lista de sesiones al recibir una nueva conexión de un cliente.	
Condiciones de ejecución: El usuario debe entrar al sistema.	
Entrada/Pasos de Ejecución: 1. Clic en la opción conectar.	
Resultado Esperado: El usuario recibe la notificación de conexión mediante el identificador de la sesión.	
Resultado Obtenido: El usuario recibe el mensaje obtenido del servidor con el identificador de la sesión.	
Evaluación de la prueba: satisfactoria	

Tabla 29: Prueba de Aceptación para la HU Adicionar sesión.

Caso de Prueba de Aceptación	
Código: H1-P2	Historia de Usuario: Eliminar sesión.
Nombre: Eliminar sesión.	
Descripción: Prueba para la funcionalidad que permite eliminar una conexión existente cuando recibe la petición de cerrar sesión.	
Condiciones de ejecución: El cliente debe cerrar sesión.	
Entrada/Pasos de Ejecución: 1. Clic en la opción desconectar.	
Resultado Esperado: El servidor elimina la comunicación con el cliente.	
Resultado Obtenido: Se elimina la sesión correspondiente a la petición recibida.	
Evaluación de la prueba: satisfactoria	

Tabla 30: Prueba de Aceptación para la HU Eliminar sesión.

Caso de Prueba de Aceptación	
Código: H1-P3	Historia de Usuario: Insertar tarea.
Nombre: Insertar tarea.	
Descripción: Prueba para la funcionalidad que permite recibir los datos a analizar a través de un mensaje desde el cliente y enviarlos a la controladora para insertarla en lista de tareas.	
Condiciones de ejecución: El cliente debe ingresar los datos a analizar.	
Entrada/Pasos de Ejecución: 1. Clic en la opción enviar tarea.	
Resultado Esperado: Se debe insertar una tarea en la lista de tareas.	
Resultado Obtenido: Se crea una tarea.	
Evaluación de la prueba: satisfactoria	

Tabla 31: Prueba de Aceptación para la HU Insertar tarea.

Caso de Prueba de Aceptación	
Código: H1-P5	Historia de Usuario: Eliminar tareas.
Nombre: Eliminar tarea.	
Descripción: Prueba para la funcionalidad que permite eliminar las tareas según son atendidas.	
Condiciones de ejecución: El servidor debe de haber atendido la tarea.	
Entrada/Pasos de Ejecución: El servidor elimina la tarea luego de ser atendida.	
Resultado Esperado: El cliente debe recibir una notificación.	
Resultado Obtenido: Se elimina la tarea de la lista de tareas.	
Evaluación de la prueba: satisfactoria	

Tabla 32: Prueba de Aceptación para la HU Eliminar tarea.

Caso de Prueba de Aceptación	
Código: H2-P1	Historia de Usuario: Crear notificación.
Nombre: Crear notificación	
Descripción: Prueba para la funcionalidad que permite crear una notificación con el estado de las tareas.	
Condiciones de ejecución: El cliente debe insertar una tarea.	
Entrada/Pasos de Ejecución: 1 El servidor le notifica al cliente cuando la tarea es enviada.	
Resultado Esperado: Se debe crear una notificación con el estado de las tareas.	
Resultado Obtenido: Se crea la notificación.	
Evaluación de la prueba: satisfactoria	

Tabla 33: Prueba de Aceptación para la HU Crear notificación.

Caso de Prueba de Aceptación	
Código: H2-P3	Historia de Usuario: Eliminar notificación.
Nombre: Eliminar notificación	
Descripción: Prueba para la funcionalidad que permite eliminar una notificación al cliente con el estado de las tareas.	
Condiciones de ejecución: El servidor debe tener al menos una notificación contenida en la lista de notificaciones.	
Entrada/Pasos de Ejecución: 1 El servidor elimina la notificación de la lista de notificaciones.	
Resultado Esperado: Se deben eliminar las notificaciones que han sido enviadas al cliente.	
Resultado Obtenido: Se elimina la notificación.	
Evaluación de la prueba: satisfactoria	

Tabla 34: Prueba de Aceptación para la HU Eliminar notificación.



Caso de Prueba de Aceptación	
Código: H3-P1	Historia de Usuario: Crear diccionario.
Nombre: Crear diccionario.	
Descripción: Prueba para la funcionalidad que permite realizar la nominalización de los datos creando una vista minable al recibirlos desde la tarea correspondiente.	
Condiciones de ejecución: La clase diccionario debe recibir los datos enviados en la tarea.	
Entrada/Pasos de Ejecución: 1 La clase Analisis recibe los datos enviados en la tarea. 2 Añade los datos a una lista y les asigna un identificador.	
Resultado Esperado: Se debe ampliar esta vista minable.	
Resultado Obtenido: Se crea una vista minable.	
Evaluación de la prueba: satisfactoria	

Tabla 35: Prueba de Aceptación para la HU Crear diccionario.

Caso de Prueba de Aceptación	
Código: H3-P3	Historia de Usuario: Ejecutar agrupamiento
Nombre: Ejecutar agrupamiento	
Descripción: Prueba para la funcionalidad que permite personalizar el ambiente de ejecución del algoritmo, introducir los datos de la vista minable y ejecutar el algoritmo de agrupamiento.	
Condiciones de ejecución: El algoritmo debe recibir los datos de la vista minable.	
Entrada/Pasos de Ejecución: 1 La clase Analisis recibe los datos. 1 Amplia la vista minable. 2 Ejecuta el agrupamiento con los datos de la vista minable.	
Resultado Esperado: Se deben obtener los grupos a partir de los datos analizados.	
Resultado Obtenido: Se ejecuta el algoritmo.	
Evaluación de la prueba: satisfactoria	

Tabla 36: Prueba de Aceptación para la HU Ejecutar agrupamiento.

Caso de Prueba de Aceptación	
Código: H3-P4	Historia de Usuario: Etiquetar grupos
Nombre: Etiquetar grupos	
Descripción: Prueba para la funcionalidad que permite como parte final del proceso de agrupamiento obtener un identificador por cada uno de los grupos el cual se asocia con cada elemento analizado dentro de la vista minable.	
Condiciones de ejecución: El algoritmo debe ejecutarse satisfactoriamente.	
Entrada/Pasos de Ejecución: 1 La clase Analisis asigna un identificador a cada grupo creado según las etiquetas definidas por el cliente.	
Resultado Esperado: Se le debe notificar al cliente que la tarea ha sido completada.	
Resultado Obtenido: Los datos son agrupados y etiquetados.	

Evaluación de la prueba: satisfactoria
--

Tabla 37: Prueba de Aceptación para la HU Etiquetar grupos.

Caso de Prueba de Aceptación	
Código: H3-P4	Historia de Usuario: Enviar resultados
Nombre: Enviar resultados	
Descripción: Prueba para la funcionalidad que permite enviar el resultado obtenido en el algoritmo de análisis al cliente.	
Condiciones de ejecución: Se debe crear un mensaje que contenga los grupos obtenidos y los identificadores asignados a estos.	
Entrada/Pasos de Ejecución: 1 El servidor le envía los resultados al cliente.	
Resultado Esperado: Se le debe notificar al cliente que la tarea ha sido completada.	
Resultado Obtenido: Los datos son agrupados y etiquetados.	
Evaluación de la prueba: satisfactoria	

Tabla 38: Prueba de Aceptación para la HU Enviar resultados.

### 3.6 Conclusiones del Capítulo

Además, se plasmaron los métodos correspondientes a la validación y prueba del modelo de Minería de Datos utilizado. Esclareciendo como se realiza el proceso de obtención del conocimiento.

Se efectuaron pruebas de aceptación al servidor mediante el prototipo de una interfaz cliente, previamente desarrollada para el sistema global desde donde se generaron las peticiones de dichas tareas.

En este capítulo se seleccionó el lenguaje de programación y el entorno de desarrollo como parte de la selección de la tecnología para dar solución al problema. Se definió como metodología de desarrollo AUP.

## **Conclusiones Generales**

Mediante esta solución se provee de un gestor de tareas que automatiza el análisis de los datos contenido en la Base de Datos de Configuración de SCADA desarrollados por el CEDIN, el cual proporciona una fuente de datos alternativa para poblar la dimensión Equipo Industrial del Almacén de Datos del SCADA, en el ambiente de Inteligencia de Negocio.

Se obtiene un sistema que significa un aumento del valor agregado a otras soluciones desarrolladas.

Se evidencia que la aplicación de técnicas y herramientas de Minería de Datos como parte auxiliar del proceso ETL hacia los Almacenes de Datos provee de nueva información que puede enriquecer el nivel de detalles en los posteriores análisis a realizar.

## RECOMENDACIONES

Para seguir aprovechando las funcionalidades de esta solución se recomienda:

La agregación de otros tipos y técnicas de análisis dentro del servidor.

Incorporar una Base de Datos para el almacenamiento del resultado de las tareas procesadas.

Implementar funcionalidades que realicen, directamente, la carga hacia el Almacén de Datos.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Belkis Grissel González, Saily Salas, Yisel Nerys Cedeño, Yusmila Vidiaux.** Propuesta de requisitos básicos para la línea de productos de software SCADA del Centro de Informática Industrial. Disponible en la web: <http://publicaciones.uci.cu/index.php/SC> | [seriecientifica@uci.cu](mailto:seriecientifica@uci.cu), 2012. Oracle Data Warehousing.
2. Inteligencia de negocio. Disponible en la web: [http://es.wikipedia.org/wiki/Inteligencia\\_empresarial](http://es.wikipedia.org/wiki/Inteligencia_empresarial), 2015.
3. **Harvey J. Miller, Jiawei Han.** Geographic Data Mining and Knowledge Discovery An Overview, 2009.
4. **MSc. Zoila Ruiz, Dr. Armando Plasencia.** MAESTRÍA CIBERNÉTICA APLICADA DIPLOMADO Gestión de Información con técnicas de Minería de Datos. ICIMAF, 2014.
5. **José Hernández Orallo y otros,** Introducción a la Minería de Dato, 2004.
6. **MSc. Zoila Ruiz, Dr. Armando Plasencia.** Árboles de Decisión. ICIMAF, 2014.
7. **Prof. Braulio José Solano.** Tareas de la minería de datos: clasificación. Universidad de Costa Rica UCR.
8. ©**Jiawei Han, Micheline Kamber.** Intelligent Database Systems Research Lab School of Computing Science Simon Fraser University, Canada. Disponible en la web: <http://www.cs.sfu.ca>
9. **José Manuel Molina, Jesús García.** TÉCNICAS DE ANÁLISIS DE DATOS. APLICACIONES PRÁCTICAS UTILIZANDO MICROSOFT EXCEL Y WEKA, 2006.
10. **MSc. Zoila Ruiz, Dr. Armando Plasencia.** Reglas de asociación. ICIMAF, 2014.
11. **García, L.** Bases de Datos y Data Warehousing, Universidad de La Habana, 2006.
12. **María Carina Roldán.** Tutorial de Pentaho Data Integration (Kettle). Disponible en la web: [http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+\(Kettle\)+Tutorial](http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+(Kettle)+Tutorial), 2009.
13. **Jorge Rivera Ramos.** Pentaho Data Integration (Kettle) parte 1 Disponible en la web: <http://mixelaneo.blogspot.com>
14. **Yrd.Doç.Dr. Ayça ÇAKMAK PEHLİVANLI.** THE COMPARISON OF DATA MINING TOOLS. Department of Computer Engineering İstanbul Kültür University submitted by Mesut ÖZKAN, 2011.
15. **WEKA.** University of Waikato. Disponible en la web: <http://www.cs.waikato.ac.nz/ml/WEKA>
16. **Prof. EMERSON CASTAÑEDA SANABRIA.** JAVA BASICO. UNIVERSIDAD CATOLICA DE COLOMBIA FACULTAD DE INGENIERIA DE SISTEMAS.
17. Visual C#. Disponible en la web: <http://msdn.microsoft.com/es-es/vcsharp/default.aspx>, 2009.
18. **Velásquez V. S.** Programación, ¿Qué clase de programas y aplicaciones se pueden crear usando C y C++? Disponible en la web: <http://viels.wordpress.com/programacion>
19. Ventajas y Desventajas: Comparación de los Lenguajes C, C++ y Java. Disponible en la web: <http://www.americati.com/doc/ventajas>, 2006.
20. **Raúl González.** Python PARA TODOS. Disponible en la web: <http://mundogeek.net/tutorial-python>
21. NetBeans. Disponible en la web: <http://www.genbetadev.com/herramientas/netbeans-1>, 2014.
22. **Shwetank Dixit.** An Introduction to WebSockets, 2012.

23. **Tamara Rodríguez Sánchez.** PROGRAMA DE MEJORA Metodología de desarrollo para la Actividad productiva de la UCI, 2014.
24. CLIENTE/SERVIDOR. SECURED.
25. JAVA - Estándares de Programación. Disponible en la web: [http://javafoundations.blogspot.com/2010/07/java-estandares-de-programacion.html#1\\_estandares](http://javafoundations.blogspot.com/2010/07/java-estandares-de-programacion.html#1_estandares)
26. Guión Visual Paradigm for UML. Disponible en la web: <http://www.ie.inf.uc3m.es/grupo/docencia/reglada/1s1y2/PracticaVP.pdf>, 2014.
27. Alejandro Esquiva Rodríguez. JSON I – ¿Qué es y para qué sirve JSON? Disponible en la web: <https://geekytheory.com/JSON-i-que-es-y-para-que-sirve-JSON/>, 2013.
28. **Joaquina Martín, Enrique Díaz.** DATA WAREHOUSE “Almacenes de Datos”, 2012.
29. **José C Riquelme, Roberto Ruiz, Karina Gilbert.** Minería de Datos: Conceptos y Tendencias. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Disponible en la web: <revista@aepia.org>, 2006
30. **José Hernández Orallo.** El Proceso de KDD. Universidad Politécnica de Valencia.
31. **Laura Reyero, Javier Peñas.** JLOP (JSON LANGUAGE ORIENTED PROCESSING). PROYECTO DE SISTEMAS INFORMÁTICOS, 2012.
32. JSON. Curso Librerías Web. Aplicaciones Web Serv Inf UA, 2008.
33. **Gonzalo Cruzado.** Base de Datos Distribuida, 2011.
34. **Inaba M, Katoh N, Imai, H.** «Applications of weighted Voronoi diagrams and randomization to variance-based  $k$ -clustering». [Proceedings of 10th ACM Symposium on Computational Geometry](#), 1994.
35. **Artefacto: Modelo conceptual.** Mejora de Procesos de Software. Disponible en la web: <http://mejoras.prod.uci.cu/>
36. **Grosso, Andrés.** Prácticas de software. Disponible en la web: <http://www.practicadesoftware.com.ar/2011/03/patrones-grasp/>, 2011.
37. **Scribd.** Diagrama de Despliegue. Disponible en la web: <http://es.scribd.com/doc/19808824/diagramas-de-despliegue-2222>.
38. Patrones GOF. Disponibles en: [http://www.ecured.cu/index.php/Patrones\\_Gof](http://www.ecured.cu/index.php/Patrones_Gof).

## BIBLIOGRAFÍA

**José Hernández Orallo y otros**, Introducción a la Minería de Dato, 2004

**Witten, I. H. & Frank, E.** Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, 2005

**Mitchell, T. M.** Machine Learning, McGraw-Hill Science/Engineering/Math, 1997

**Han, J.** Data Mining Concepts and Techniques, Morgan Kaufmann, 2006

**Xu, R. & Wunsch, D. C.** Clustering, IEEE Press, 2007

**Césari Matilde.** Aprendizaje automático con WEKA.

Inteligencia de negocio. Disponible en la web: [http://es.wikipedia.org/wiki/Inteligencia\\_empresarial](http://es.wikipedia.org/wiki/Inteligencia_empresarial), 2015.

Historias de usuario. Disponible en la web: [http://es.wikipedia.org/wiki/Historias\\_de\\_usuario](http://es.wikipedia.org/wiki/Historias_de_usuario), 2015.

Patrones de Diseño.

[https://www.google.com.cu/?gws\\_rd=cr,ssl&ei=LZaKVanRE8H7-AGZko3lCA#q=api+de+java](https://www.google.com.cu/?gws_rd=cr,ssl&ei=LZaKVanRE8H7-AGZko3lCA#q=api+de+java)

**Maria-Isabel Sánchez, Arturo Mora.** Diagramas de colaboración, Metodología de desarrollo visual, Universidad Carlos III de Madrid.

Diagramas de Paquetes. Disponible en la web: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_paquetes,2015](http://es.wikipedia.org/wiki/Diagrama_de_paquetes,2015).

Como agrupar objetos de un misma categoría. Análisis de Agrupamiento. Disponible en la web: <https://www.statsoft.com/Textbook/Cluster-Analysis#two>.

¿Qué significa http? - Definición de http. Disponible en la web: <http://www.masadelante.com/faqs/que-significa-http>.

## Glosario de Términos

**SCADA:** acrónimo de Supervisory Control And Data Acquisition (en español Supervisión, Control y Adquisición de Datos) es un software para ordenadores que permite controlar y supervisar procesos industriales a distancia.

**CEDIN:** acrónimo de Centro de Informática Industrial, centro de desarrollo de aplicaciones industriales perteneciente a la Universidad de las Ciencias Informáticas.

**Norma:** Documento, establecido por consenso y aprobado por un organismo reconocido, que proporciona, para uso común y repetido, reglas, directrices o características para las actividades o sus resultados, destinado al logro de un grado óptimo de orden en un contexto dado. ONN (Oficina Nacional de Normalización).

**Equipo Industrial:** Es una dimensión del Almacén de Datos de los SCADA del CEDIN dedicada a la persistencia de los datos referentes a los equipos que se usan en el proceso de producción de las industrias.

**Almacén de Datos:** Un Almacén de Datos es una colección de datos orientado a temas o materias, integrado, no volátil y variable en el tiempo, que será utilizado fundamentalmente en el proceso de toma de decisiones.

**ETL:** acrónimo de Extract, Transform and Load (en español Extraer, Transformar y Cargar) es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

**Simple K-means:** Es el algoritmo de agrupamiento más conocido y se encuentra clasificado como un algoritmo particional. Está basado en la minimización de la distancia interna entre los elementos de los K grupos definidos por el usuario que realiza la minería.

**Minería de Datos:** Se define a la Minería de Datos como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos.

Inteligencia Artificial: Programa de computación diseñado para realizar determinadas operaciones que se consideran propias de la inteligencia humana, como el autoaprendizaje.

**BI:** acrónimo de Business Intelligence (en español Inteligencia de Negocios), es la combinación de tecnología, herramientas y procesos que permiten transformar los datos almacenados en información, esta información en conocimiento y este conocimiento dirigido a un plan o una estrategia comercial.

**KDD:** acrónimo de Knowledge Discovery from Databases (en español descubrimiento de conocimiento de las bases de datos) es una contestación a los enormes volúmenes de datos que son reunidos y almacenados en las bases de datos operacionales y científicas.

**IT:** acrónimo de Information Technology (en español Tecnología de Información) es el proceso de mejora en la tecnología de la información.

**ARFF:** acrónimo de Attribute-Relation File Format (en español Archivo de Formato Atributo- Relación), archivo soportado por WEKA.

**IETF:** acrónimo de Internet Engineering Task Force (en español, Grupo de Trabajo de Ingeniería de Internet) para promover, a nivel global, la participación en actividades que ayuden a conservar la integridad de Internet.

**C4.5:** Es la extensión del algoritmo ID3 para la generación de árboles de decisión en el agrupamiento.

**ID3:** El algoritmo ID3 es utilizado dentro del ámbito de la inteligencia artificial. Su uso se engloba en la búsqueda de hipótesis o reglas en él, dado un conjunto de ejemplos.

**API:** La API Java es una interfaz de programación de aplicaciones (API, por sus siglas del inglés: Application Programming Interface) provista por los creadores del lenguaje de programación Java, que da a los programadores los medios para desarrollar aplicaciones Java.

**ANSI:** es un estándar publicado por el Instituto Nacional Estadounidense de Estándares (ANSI), para el lenguaje de programación C. Se recomienda a los desarrolladores de software en C que cumplan con los requisitos descritos en el documento para facilitar así la portabilidad del código.

**HTTP:** acrónimo de Hipertexto Transfer Protocol (en español, Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la world wide web, el método mediante el cual se transfieren las páginas web a un ordenador.

**W3C:** acrónimo de **World Wide Web Consortium**, es una comunidad internacional donde la organización miembro, un equipo de trabajo a tiempo completo, y el público, trabajan de conjunto para desarrollar estándares web.

**GRASP:** acrónimo de General Responsibility Assignment Software Patterns (en español, Patrones de Software para la Asignación General de Responsabilidad).

**GOF:** acrónimo de Gang-of-Four (en español, "pandilla de los cuatro") se descubren como una forma indispensable de enfrentarse a la programación a raíz del libro "Design Patterns—Elements of Reusable Software" de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides, a partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro (GoF, gang of four).



## Anexos


Historia de Usuario	
<b>Número: 4</b>	<b>Nombre del requisito: Modificar tareas</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 1</b>
<b>Prioridad: Media</b>	<b>Tiempo Estimado:4h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 3h</b>
<b>Descripción:</b> Permite a la controladora modificar el contenido de una tarea existente en la lista de tareas una vez que el cliente modifica una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	
	

Tabla 39: Historia de usuario número 4. Modificar tareas.

Historia de Usuario	
<b>Número:7</b>	<b>Nombre del requisito: Enviar notificaciones</b>
<b>Programador: Deyanira Bonaparte Pérez.</b>	<b>Iteración Asignada: 2</b>
<b>Prioridad: Alta</b>	<b>Tiempo Estimado:48h</b>
<b>Riesgo en Desarrollo:</b> Problemas técnicos y eléctricos.	<b>Tiempo Real: 44h</b>
<b>Descripción:</b> Permite enviar al cliente correspondiente el estado de las tareas a través de notificaciones.	

**Observaciones:**

**Prototipo de interfaz:**

LOGS DESDE EL SERVER

039e7f5e-9000-4cbd-a9d0-120b9cb043b8 => Session Opened  
2c45ad04-0ac0-4a6b-922e-1297a7636bb4 => Session Opened  
Message for task received => PENDIENTE  
Message for task received => DONE!  
Message for task received => PENDIENTE  
Message for task received => DONE!

Tabla 40: Historia de usuario número 7. Enviar notificaciones.

**Plan de prueba para la Historia de Usuario 4: Modificar Tarea**

Descripción General: Se encarga de Modificar una tarea en el gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe seleccionar la opción Modificar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU4: Modificar Tarea	Escenario 4.1: Modificación de tarea satisfactoria.	Se modifica el nombre de la tarea, las diferentes etiquetas y el contenido para analizar y se envía al servidor de análisis.	1 Seleccionar la opción Modificar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client. 2 En la ventana EDITAR DATOS DE LA TAREA modificar nombre de la tarea, las diferentes etiquetas y el contenido para analizar. 3 Seleccionar la opción Enviar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client. 4 En la ventana LOGS DESDE EL SERVER una vez que la tarea se ha modificado aparece la notificación Pendiente
	Escenario 4.2: Datos en blanco.	En caso de modificar el contenido de la tarea y dejarlo en blanco.	1 Seleccionar la opción Modificar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client. 2 En la ventana

			<p>EDITAR DATOS DE LA TAREA modificar nombre de la tarea y las diferentes etiquetas.</p> <p>3 Seleccionar la opción Enviar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.</p> <p>4 En la ventana LOGS DESDE EL SERVER aparece la notificación BAD indicando que la tarea no se ha modificado por algún error.</p>
--	--	--	---

Tabla 41: Plan de prueba para la Historia de Usuario 4. Modificar Tarea.

### Diseño de caso de prueba 4: Modificar Tarea.

C: Datos correctos, V: Campo vacío.

Id de la sección	Esce nario	Variable 1 Nombre de Tarea	Variable 2 Contenido	Variable 3 Etiquetas	Respues ta del sistema	Resultado de la prueba
P.HU4	Modificar Tarea	c	c	c	Se modifica la tarea	Prueba satisfactoria
		v	c	c	Se modifica la tarea	
		c	c	v	Se modifica la tarea	
		c	v	c	No se modifica la tarea	
		c	v	c	Se modifica la tarea	Prueba no satisfactoria

Tabla 42: Diseño de caso de prueba 4. Modificar Tarea.

### Plan de prueba para la Historia de Usuario 7: Enviar notificaciones

Descripción General: Se encarga de Enviar notificaciones en el gestor de tareas de análisis.

Condiciones de Ejecución: El servidor debe contar con al menos una notificación en la lista de notificaciones.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU7: Enviar notificaciones	Escenario 7.1: Envío de notificaciones satisfactorio.	Se envían notificaciones del servidor de análisis al cliente.	1 El servidor chequea si para cada sesión iniciada existen notificaciones pendientes por enviar. 2 Envía la notificación a la sesión correspondiente.
	Escenario 7.2: Envío de notificaciones insatisfactorio.	En caso de que el servidor presente fallas y deje de funcionar.	1. Durante el funcionamiento en el servidor se produce una excepción debido a un desbordamiento de memoria.

Tabla 43: Plan de prueba para la Historia de Usuario 7: Enviar Notificaciones.

### Diseño de caso de prueba 7: Enviar Notificaciones.

A: botón accionado NA: botón no accionado

Id de la sección	Escenario	Variable 1 Botón Conectar al servidor	Variable 2 Botón Desconectar	Variable 3 Eliminar tarea	Respuesta del sistema	Resultado de la prueba
P.HU7	Enviar Notificaciones	A	A	NA	Se envía la notificación	Prueba satisfactoria
		NA	NA	A	No se envía la notificación	
		NA	NA	A	Se envía la notificación	Prueba no satisfactoria

Tabla 44: Diseño de caso de prueba 7. Enviar Notificaciones.

Caso de Prueba de Aceptación	
Código: H1-P4	Historia de Usuario: Modificar tareas.
Nombre: Modificar Tarea.	
Descripción: Prueba para la funcionalidad que permite modificar una tarea creada y enviarla al servidor de análisis.	
Condiciones de ejecución: La tarea debe estar creada.	

Entrada/Pasos de Ejecución: 1 El servidor recibe una tarea. 2 Inserta la tarea en la listas de tareas. 3 Envía una notificación al cliente.
Resultado Esperado: Se debe enviar una notificación al cliente.
Resultado Obtenido: Se inserta una tarea en la lista de tareas.
Evaluación de la prueba: satisfactoria

Tabla 45: Prueba de Aceptación para la HU Modificar tareas.

Caso de Prueba de Aceptación	
Código: H2-P2	Historia de Usuario: Enviar notificación.
Nombre: Enviar notificación	
Descripción: Prueba para la funcionalidad que permite enviar una notificación al cliente con el estado de las tareas.	
Condiciones de ejecución: El servidor debe tener al menos una notificación contenida en la lista de notificaciones.	
Entrada/Pasos de Ejecución: 1 El servidor le notifica al cliente el estado de la notificación.	
Resultado Esperado: Se deben obtener el estado de las notificaciones.	
Resultado Obtenido: Se envía la notificación.	
Evaluación de la prueba: satisfactoria	

Tabla 46: Prueba de Aceptación para la HU Enviar notificación.

## Plan de prueba para la Historia de Usuario 10: Ampliar el ámbito de análisis

Descripción General: Se encarga de Ampliar el ámbito de análisis en el gestor de tareas de análisis.

Condiciones de Ejecución: El cliente debe seleccionar la opción: Agregar Etiquetas en la parte EDITAR DATOS DE LA TAREA de la interfaz principal BISA Smart Client.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
P.HU10: Ampliar el ámbito de análisis	Escenario 10.1: Amplía el ámbito de análisis satisfactoriamente.	Se envían las etiquetas ingresadas por el cliente al servidor para ampliar el ámbito de análisis.	1 El servidor utiliza las etiquetas que recibe desde el cliente para ampliar el diccionario de la vista minable. 2 Se establecen los pesos de los valores de las etiquetas dentro de la vista minable.
	Escenario 10.2: No	En caso de no	1 Seleccionar la opción Agregar o Modificar

	amplía el ámbito de análisis.	especificar las etiquetas.	Tarea en la parte superior izquierda de la interfaz principal BISA Smart Client. 2 En la ventana EDITAR DATOS DE LA TAREA insertar nombre de la tarea y el contenido para analizar, sin insertar las etiquetas. 3 Seleccionar la opción Enviar Tarea en la parte superior derecha de la interfaz principal BISA Smart Client.
--	-------------------------------	----------------------------	---

Tabla 47: Plan de prueba para la Historia de Usuario 10. Ampliar el ámbito de análisis.

### Diseño de caso de prueba 10: Ampliar el ámbito de análisis.

C: Datos correctos, V: Campo vacío.

Id de la sección	Esce nario	Variable 1 Nombre de Tarea	Variable 2 Contenido	Variable 3 Etiquetas	Respues ta del sistema	Resultado de la prueba
P.HU10	Ampliar el ámbito de análisis	c	c	c	Se amplía el ámbito de análisis.	Prueba satisfactoria
		v	c	c	Se amplía el ámbito de análisis.	
		c	c	v	No se amplía el ámbito de análisis.	
		c	c	c	No se amplía el ámbito de análisis.	Prueba no satisfactoria

Tabla 48: Diseño de caso de prueba 10. Ampliar el ámbito de análisis.

Historia de Usuario	
Número: 2	Nombre del requisito: Eliminar sesión
Programador: Deyanira Bonaparte Pérez.	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 24h
Riesgo en Desarrollo: Problemas técnicos y eléctricos.	Tiempo Real: 10h

**Descripción:** Permite al servidor eliminar una conexión existente cuando recibe la petición de cerrar sesión.

**Observaciones:**

**Prototipo de interfaz:**

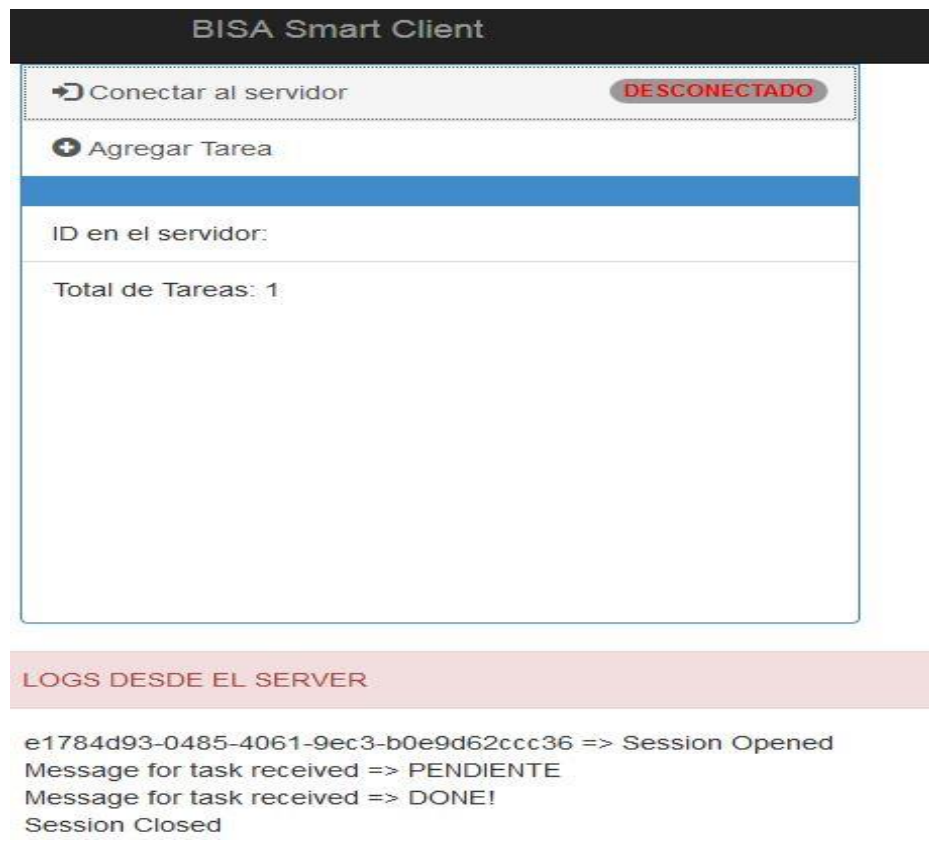


Tabla 49: Historia de usuario número 2. Eliminar sesión.