SOFTWARE PARA APOYAR LA TOMA DE DECISIONES EN EL AREA DE MERCADEO Y VENTAS

Grupo de Investigación:

E-Soluciones

Línea de Investigación:

E-Servicios

Director del proyecto:

Prof. Plinio Puello Marrugo

Investigadores:

José Luis Bolaños Herazo

Julio Cesar Suarez Carrillo

UNIVERSIDAD DE CARTAGENA

PROGRAMA INGENIERIA DE SISTEMAS

CARTAGENA DE INDIAS DT Y C

2011

NOTA DE ACEPTACIÓN

_____

_____

_____

_____

_____

_____

_____

Firma del Presidente del Jurado

_____

Firma del Jurado

_____

Firma del Jurado

Cartagena de Indias,2011

CONTENIDO

Índice de Gráficas

Índice de Tablas

RESUMEN

El presente proyecto se desarrolló por la motivación de implementar un sistema software capaz de extraer información relevante a partir de las ventas de los productos de un negocio en un periodo de tiempo de forma automática, y permitir que esta información sea mostrara al usuario quien lo utilizaría para decidir objetivamente que productos comercializar para estar acorde a la necesidad de los consumidores. Para lograr este propósito, el sistema almacena los productos, registra las compras, y utiliza una versión modificada del algoritmo Apriori para obtener las tendencias de compra de los productos.

Se probó el sistema con los datos de las ventas una empresa local, con base en esto se obtuvieron la tendencia de compra de los clientes y afinidad de compra de un conjunto de productos. El diseño de la aplicación se centro en la capacidad de ser accesible desde la web y dispositivos móviles y con soporte para distintitas bases de datos como: MYSQL, SQLITE, POSGRESQL, MSSQL, ORACLE, entre otras.

Se concluyó que la aplicación del sistema software de apoyo a las decisiones obtuvo un panorama favorable, puesto que el algoritmo Apriori arroja resultados precisos sincronizándose con los datos de las ventas que almacena la aplicación web. También se evidenció que la modificación de Apriori fue más rápida que la versión original.

Palabras claves: algoritmo Apriori, decisiones, productos, tendencia de compras, ventas.

ABSTRACT

In this project was developed an implementation of a system with the capability of getting accurate information over the sales of a market or minimarket for a period of time, and all this automatically; besides, in order to take the best decision commercializing the products without ignoring the trends of the consumers, the mentioned information is shown to the end user through a friendly interface. The system store products, register transactions as well, and use a modified Apriori algorithm in order to get sales trends in a bunch of products.

The system was tested with real sale data of local market, the result was the products sales trends and affinity of the costumers. The system support variety of database and his architecture is on web technology, in order to have the capability of being accessible from everywhere, and has a version to mobile devices.

As research results, the software got good ones: first of all the modified algorithm was faster than the original one; second, the synchronization between the modified Apriori and the web application was a success; and finally, the web implementation shown than not only give an accessibility from everywhere but allows use it in mobile devices whose tech power is low.

Palabras claves: algoritmo Apriori, decisiones, productos, tendencia de compras, ventas

Keywords: Apriori algorithm, decisions, products, sales, sale trends

1 INTRODUCCIÓN

Las empresas tienen la necesidad de analizar la información a favor de tomar decisiones que a corto plazo apoyen su gestión y competencia en el mercado. De aquí es donde surge la necesidad de incorporar una herramienta de software que extraiga los datos relevantes de un volumen grande de información para apoyar las decisiones de su patrimonio.

La información debe responderle a los socios inquietudes como: ¿Qué compran los clientes?, ¿Qué patrones siguen para comprarlo? Las respuestas a estas y otras preguntas están en los registros de compras de los clientes, los registros de artículos, y la cantidad de ventas en un periodo, los cuales dentro de esta investigación se tuvieron en cuenta para el desarrollo de un sistema de apoyo a las decisiones. Este sistema software de apoyo a las decisiones será aplicado a análisis de ventas dando como resultado una reglas de decisión que permitirán la construcción de tendencias, gráficos, predicciones de productos vendidos, control de inventario, relaciones. También se tendrán en cuenta el comportamiento de las características de los productos que más influyen en una transacción buscando identificación de los patrones de venta

Este sistema se desarrollo basado en el algoritmo de minería de datos Apriori, el cual es capaz de procesar una gran cantidad de datos y extraer conjuntos de datos frecuentes, basados en parámetros de confiabilidad y un soporte, obteniendo un índice más confiable de los resultados.

Para realizar este sistema se baso en la metodología CRIPS-DM, esta permitió identificar características de ventas a analizar, estudiar los algoritmos de minería de datos, desarrollar una mejora del algoritmo estudiado, implementar el sistema de apoyo de decisiones en plataformas web y móvil. Y también verificar su funcionamiento con los datos de ventas de una empresa local.

El desarrollo del software va a seguir un patrón de diseño, que permite ser modificable y mantenido con mayor comodidad por terceros para contribuir a la implementación de futuras mejoras, el modelo elegido fue el Modelo Vista Controlador (MVC). La

documentación y el manual de usuario van a ser agregados para que terceros, en especial usuarios, comprendan el funcionamiento.

2 OBJETIVOS

2.1 Objetivo General

Diseñar y desarrollar un software para apoyar la toma de decisiones en el área de mercadeo y ventas con una aplicación web haciendo uso de los conocimientos de algoritmos de asociación.

2.2 Objetivos Específicos

- Estudiar los algoritmos de asociación utilizados en la literatura.
- Realizar un nuevo algoritmo a partir del algoritmo Apriori.
- Comparar con diferentes valores de soporte y confianza esta implementación con la tradicional
- Diseñar un software de manejo de productos
- Adaptar el algoritmo realizado al software de manejo de productos en plataforma web.
- Mostrar la viabilidad de adaptar un algoritmo asociación a una aplicación web

3 ALCANCE

Cumpliendo el propósito del proyecto desarrollar un software que ayude a la toma de decisiones con respecto a los productos que vende una pequeña o mediana empresa, para ello se hace uso del conocimiento de algoritmos de asociación y se utilizan datos reales de ventas proporcionados por un negocio del sector. Los pasos que abarcó el proyecto fueron los siguientes:

- La modificación al algoritmos Apriori
- La comparación el rendimiento del algoritmo Apriori y la modificación
- La construcción de una aplicación web que almacene productos y las ventas que se producen
- La integración de la modificación del algoritmo con la aplicación web.
- La ejecución del software con datos reales de venta.

4 DESCRIPCIÓN DEL PROYECTO

4.1 PROBLEMA DE INVESTIGACIÓN

En todas las empresas los sistemas de información son esenciales para el desarrollo a nivel interno y externo, lo que conlleva al mejoramiento de procesos de producción. La información dentro de la empresa sobre productos, balance de costos, ventas, inventario, características de usuarios son las que apoyan a cada área funcional en la generación de estrategias para alcanzar los objetivos de la empresa.

La manera como se manipula la información dentro de dichas organizaciones se puede presentar de forma manual o sistematizada. De acuerdo con ello, las empresas que basan sus estrategias sin utilizar sistemas de gestión poseen desventajas frente a aquellas que realizan estos análisis de forma sistemática; entre estas se destacan: el tiempo de análisis, las decisiones poco confiables y énfasis en experiencia por parte de asociados para resolver problemas.

La inclusión de sistemas de gestión dentro de las compañías han mejorado los procesos organizacionales dándole un mejor control a nivel interno sobre la producción y ganancias. Empleando dichos sistemas en áreas más especificas como son: ventas, marketing, suministros, entre otros, permiten definir estrategias de comercialización que influyan al mejoramiento, crecimiento y expansión de la organización, esto es lo que se le conoce como un sistema de apoyo para la toma de decisiones.

Sin embargo, a pesar de las ventajas de estos sistemas, de acuerdo al estudio de "Aproximación al proceso de toma de decisiones en la empresa barranquillera" (Cabeza de Vergara & Muñoz Santiago, 2004) en los que tomaron como muestra a 77 empresas barranquilleras, reflejó que el 40% de ellas no usaban herramientas computacionales para tomar decisiones y lo compensaban con la intuición. Esto se debe, según el análisis del mismo trabajo, a la poca oferta de programas, pensado en el mercado colombiano, que analicen grandes cantidades de datos en un área de la empresa y genere información útil para tomar las decisiones en dicha área.

En el caso del área de mercadeo los sistemas deben ser capaces de analizar las transacciones internas que afectan las ventas, de disponer de la información pertinente al producto o los productos, de mantener información actualizada de las transacciones externas que influyen en los productos que circulan en el mercado, de estudiar si sus características pueden ser cambiadas o mejoradas. Al combinar estos factores se alcanzaría mejorar la forma como se organiza la información para garantizar el desarrollo, mejoramiento y fortalecimiento en el mercado.

Herramientas existentes como Trident, manejan el inventario y las ventas, además pueden hacer cálculos contables, que son útiles para dar reportes a entidades como la DIAN. Pero no pueden extraer cómo se comportan sus ventas a través de los clientes. Software como estos son construidos por la demanda de las necesidades de grandes empresas de grandes cálculos contables pero para una empresa pequeña de venta de mercancía necesita también conocer cómo se comportan sus ventas. El software a desarrollar tendrá la capacidad de almacenar las ventas de los productos y arrojar el comportamiento de las mismas de forma automática.

Con base en lo anterior, para mejorar estos sistemas de soporte a las decisiones en el software a desarrollar se incluirá métodos de análisis basado en algoritmo de asociación. Apriori introducido por Agrawal descrito en "Mining Association Rules between Sets of Items in Large Databases" (Agrawal, Tomasz, & Swami, 1993). Aplicado a las ventas permite encontrar que producto o conjunto de productos conlleva a que otro conjunto sea comprado. Mejor conocido a estos conjuntos antecedentes y consecuentes respectivamente. La literatura nombra otros algoritmos de asociación como son FP-Growth descrito en los trabajos "Mining frequent patterns without candidate generation" (Han, Pei, & Yin, 2000), "An Implementation of the FP-growth Algorithm" (Borgelt, 2005), y el algoritmo de Eclat descrito en "Efficient Implementations of Apriori and Eclat y Eclat: Automatic generation and classification of test inputs" (Borgelt, 2003a; Pacheco & Ernst, 2005), que utilizan arboles de prefijos para obtener de manera rápida los conjuntos de ítems frecuentes, pero con dicho árbol no es posible obtener reglas de asociación y es necesario recorrer las transacciones tal cual como lo formula el algoritmo de Apriori, razón por la cual los autores omiten este término en sus trabajos. Además las implementaciones en estos trabajos son estáticas, lo que significa

que los algoritmos deben recorrer nuevamente todo el dataset incluso cuando este es una actualización de nuevas transacciones. Esta podría ser la razón por la que estas implementaciones no se combinan con alguna aplicación para obtener los conjuntos de datos y obtener reglas de asociación automáticamente.

Con base a estas observaciones se propone una variación del algoritmo Apriori en donde se mejore su rendimiento marcando las transacciones que ya no poseen ítems frecuentes con el fin de no recorrer el total de las transacciones cada vez que se necesita verificar el soporte de un conjunto de datos candidatos. Una vez que se verifica el soporte de un conjunto de ítems candidatos se marcará aquellas transacciones que no aportaron información para el cálculo del soporte de los nuevos conjuntos de datos venideros. También se evita crear una nueva estructura de transacciones cada vez que se generan nuevos conjuntos de ítems como ocurre en AprioriTid postulado por (Agrawal & Srikant, 1994).

Además de usar el formato establecido en que los datos son tomados de un archivo de texto en la que cada fila es una transacción y los ítems están separados por espacios, se sugiere cambiarlo por un lenguaje de serialización como es YAML (pronunciado fonéticamente "yæməl") descrito en "YAML Ain't Markup Language (YAML™) Version 1.2." (Ben-KiKi, Evans, & döt Net, s.d.), que es fácilmente legible para las personas y basado en una estructura más fácil de extraer de bases de datos, para ser procesada por el algoritmo, que consiste en que cada registro represente el id de la transacción y un producto asociado como fue realizado en los experimentos con base de datos relacionales en (Sarawagi, Thomas, & Agrawal, 2000). Este formato será el empleado para crear el archivo con los datos de transacciones en tiempo de ejecución para ser minado con el algoritmo de asociación descrito.

Los trabajos previos con estos algoritmos se centran en hacerlos estáticos, lo que significa que al insertar nuevas transacciones en el dataset es necesario realizar el proceso del algoritmo nuevamente sobre todas las transacciones. En esta implementación se utilizará una heurística en que se pueden calcular nuevos ítems frecuentes y sus respectivas reglas sin ser imperativo recorrer todas las transacciones de nuevo. Esta característica será útil cuando se una a la aplicación web que controla la entrada de productos.

El desarrollo de esta implementación de Apriori será comparada con el tradicional postulado en "Mining Association Rules between Sets of Items in Large Databases" (Agrawal et al., 1993), se medirá el comportamiento con diferentes valores de soporte y de confianza, para ello se utilizarán datos suministrados por *PARTY TIMES & DEKO LTDA*. Posteriormente a las pruebas, para verificar el dinamismo del algoritmo desarrollado a la entrada de nuevas transacciones se desarrollará un software web que permita registrar productos y la venta de los mismos.

Con el fin de usar datos de prueba, la empresa *PARTY TIMES & DEKO LTDA* servirá de modelo para la realización del software. *Party Times & DEKO LTDA* es una empresa dedicada a la compra y venta de juguetería y piñatería. Su misión se encamina hacia: "Ofrecer a nuestros clientes los productos y servicios de la más alta calidad, al precio justo, en el ámbito adecuado, procurando su más amplia satisfacción a través de un esmerado servicio personalizado. El cliente es la razón de ser de nuestro trabajo". Y su visión es: "Consolidar y mantener el liderazgo de nuestra Empresa en el mercado, integrando los objetivos de sus clientes, personal, proveedores y accionistas".

Los datos proveídos son los artículos que se venden agrupados por sus secciones, y las ventas de ellos en los últimos meses. Estos permiten hacer pruebas unitarias y funcionales al software que se desarrollará. Con la información de las ventas se verificará el correcto funcionamiento del algoritmo a desarrollar, y posterior a eso se incluirá en el software encargado de las ventas de productos, cuyo objetivo es que sea de utilidad a todas aquellas microempresas que venden productos.

Debido a que en la actualidad se busca que la información esté disponible desde cualquier lugar, este prototipo no se desarrollará bajo tecnologías de escritorio. Las aplicaciones de escritorios son veloces pero carecen de la movilidad, prescindiendo completamente de ella en ocasiones importantes, como por ejemplo comprando mercancía o haciendo negocios en otro país. Esta barrera es rota con una aplicación web, pero se debe tener en cuenta que son frágiles si se desarrolla con poco conocimiento de seguridad.

Las pruebas se realizarán al proyecto en su parte lógica, visual y de seguridad. Se pretende utilizar diferentes herramientas que sean necesarias para garantizar la calidad

del software (aplicación web y algoritmos). Así se utilizará en lo posible herramientas de automatización de pruebas a nivel de código (caja blanca) y de resultados, y herramientas que midan la seguridad en la parte lógica, en conjunto se utilizarán herramientas de penetración y de testeo de vulnerabilidades.

4.2 FORMULACIÓN DEL PROBLEMA

¿Cómo mejorar la toma de decisiones en una empresa a partir de sus ventas? ¿Es factible un software que analice los datos de las ventas con el fin de almacenar y obtener información útil a disposición de una empresa? ¿Es posible realizar una adaptación mejorada del algoritmo Apriori en una aplicación web que maneja las ventas?

4.3 JUSTIFICACION

Actualmente, muchas organizaciones manejan de forma manual o con herramientas básicas la información correspondiente a las variables del mercado que afectan los productos y los servicios que ofrecen. Es por ello que se considera fundamental apoyar la toma de decisiones de las empresas con base en software especializados que permitan un manejo apropiado de la información y que garanticen la relación e interacción de las tendencias del mercado y fortalezca la toma de decisiones para mantener la competitividad de las empresas a través del fortalecimiento de la posición de los productos en el mismo.

La manera apropiada para alcanzar el objetivo propuesto se argumenta en el hecho de que a través del software es posible alcanzar y lograr de manera objetiva el análisis de las transacciones internas que afectan la ventas, lo anterior elimina la subjetividad que podría presentarse en la toma de decisiones y que, en muchas ocasiones, ha favorecido para aceptar ideas erradas en los negocios.

Este proyecto nace con el fin de realizar un software que contraste con aquellos que se venden en el mercado a nivel regional. El funcionamiento de estos se basa en arrojar

resultados contables del flujo de caja, pero adolecen de poseer funciones que descubran el comportamiento que poseen las mercancías al ser vendidas que se traducen en los hábitos de compras de los clientes.

También el proyecto se empeña en integrar los conceptos de los algoritmos de asociación en un software que almacena las transacciones de ventas en una base de datos. A diferencia de proyectos "A fast APRIORI implementation. In Proceedings Of The Ieee Icdm Workshop On Frequent Itemset Mining Implementations" (Bodon, 2003), "Efficient Implementations of Apriori and Ecla" (Borgelt, 2003a), "Recursion Pruning for the Apriori Algorithm" (Borgelt, 2003b), "An Implementation of the FP-growth Algorithm" (Borgelt, 2005), "Efficiently Using Prefix-trees in Mining Frequent Itemsets" (Grahne & Zhu, 2003), "LCM: An efficient algorithm for enumerating frequent closed item sets" (Uno, Asai, Uchida, & Arimura, 2003). En los cuales hacen pruebas a los algoritmos de asociación que son desarrollados independientemente, de forma estática y haciendo mediciones solo variando el soporte.

Por otra parte, otras herramientas poseen demasiadas funcionalidades que para las ventas de productos no son necesarias y por ello el costo y el manejo de recursos computacionales son demasiado elevados. Además estos software son de escritorio y no se puede acceder desde lugares distintos a la red donde se encuentra instalado (si la hay) en cambio el prototipo a desarrollar podrá ser accedido desde cualquier lugar con conexión a internet a través de una interfaz amigable y aplicando procedimientos de seguridad.

La realización se llevará a cabo con el uso de los conocimientos obtenidos en lo largo de las asignaturas de la carrera ingeniería de sistemas, como son base de datos, ingeniería de software, seguridad informática, inteligencia artificial, sistemas de información gerencial, entre otros. Las herramientas de hardware y software son brindadas por la universidad y la información pertinente como son los datos e información de la investigación es obtenida gracias a la microempresa. Se utilizará lenguajes de programación de distribución libre como *Java*, *PHP* e IDE'S como *Netbeans* o *PHP EDITOR*. Con el software se contribuirá a la sociedad con una herramienta de ventas pensada para las pequeñas comercializadoras de productos que contribuye a su vez a mostrar como el uso de las ciencias de la computación ayuda en esta área.

La aplicación será desarrollada utilizando el modelo de desarrollo en espiral que permite adaptar de manera rápida los requerimientos de los usuarios a los que va dirigido. Se plantea realizar pruebas funcionales, unitarias, y de seguridad para garantizar la calidad del software. Este procedimiento de desarrollo permitirá ahorrar tiempo, esfuerzo y brindar buenos resultados a la vez.

Por otra parte, el software aseguraría la disposición de la información pertinente al producto o servicio, con esta ventaja se garantizaría la inclusión de todas y cada unas de las variables que intervienen en el mercado sin omitir dato alguno, para poder así tener en cuenta los elementos que en un momento dado afectarían el desarrollo, mejoramiento y fortalecimiento deseado. De igual manera, el hecho mantener la información actualizada de las transacciones externas/internas que se relacionan con los productos/servicios que se ofertan y, al estudiarlos permite que las ya mencionadas características y elementos de identificación de los mismos puedan ser cambiados o mejorados de acuerdo a los requerimientos que se presenten. De ahí que, al combinar estos factores se alcanzaría mejorar en la organización de la información apoyando la competitividad de las empresas.

De otro lado, la realización de este proyecto contribuye al afianzamiento de los conocimientos adquiridos durante los estudios de Ingeniería de Sistemas. Algunas de las asignaturas relacionadas con la temática de este proyecto son: programación, metodología de la investigación, seguridad de la información administración, simulación, entre otros. Siendo un aporte para el componente profesional complementario que quieren adquirir los egresados del Programa de Ingeniería de Sistemas, en concordancia con el proyecto educativo del programa, la misión y visión del mismo. La investigación se orienta al desarrollo de un producto que ofrece servicios a nivel empresarial usando conceptos de los sistemas de información gerencial conceptos de minería de datos, técnicas de simulación e inteligencia artificial, los cuales son incluidos en la línea de investigación de E-Servicios. Dentro de la línea de investigación se fortalecerá el estudio realizado con los conocimientos del grupo sobre las diferentes técnicas de manejo y extracción de información así como sus técnicas usadas para las simulaciones y se buscara que conocimientos adquiridos durante el transcurso de la investigación sirvan de ayuda para futuros proyectos de software. .

Finalmente, este aplicativo será para las empresas una herramienta para estudiar y analizar los comportamientos de los productos (Oferta vs. Demanda) en el mercado, apoyando la toma de decisiones desde el área administrativa de las organizaciones. Las funcionalidades que ofrecerá el software son las siguientes: Procesamiento de datos, generación de estadísticas, reportes de comportamientos de los productos, Ingreso, Actualización, consulta de información y por su puesto información del comportamiento de las compras a través del desarrollo del algoritmo propuesto.

5 ESTADO DEL ARTE Y MARCO TEÓRICO

Las empresas registran sus ventas en sistemas de base de datos, permitiendo explotarla realizando un análisis para encontrar asociaciones interesantes en la compra de productos para la toma de decisiones en el proceso de mercadeo y ventas.

Las herramientas como Trident («Software Contable Trident Enterprise», s.d.) O SAP («SAP Andeancarib - SAP Professionals - ¿Qué son los módulos SAP?», s.d.) Son utilizadas para manejar la información de ventas de producto. Estas realizan un análisis mediante un escaneo a la base de datos en el que se busca encontrar hechos que ocurren con mayor frecuencia.

Este estudio de la información se realiza a través de reglas de asociación incluidas en "Minería de Reglas de Asociación con Programación Genética" (Luna, Olmo, Romero, & Ventura, 2010) y aprendizaje automático, donde la forma de obtener información a través de los datos con propósitos comerciales se le conoce como Business Intelligent.

5.1 Business Intelligent

Business Intelligent es el conjunto de metodología, tecnologías y aplicaciones que permiten reunir y transformar datos de los sistemas de transacción e información desestructurada en información para su análisis y conversión en conocimiento, y de esta forma dar soporte a la toma de decisiones de la empresa.

Business Intelligent (BI) funciona como un factor de estrategia de una empresa, para aumentar su capacidad competitiva en el mercado, que no es otra que poseer la información privilegiada para responder a los problemas de un negocio como son: entrada a nuevos mercados sin explorar, ofertas de productos, control financiero, reducción de los costos de producción, planificación de la producción, los perfiles de clientes (gustos), rentabilidad de un producto concreto, entre otros (Gaik Yee, Aziz, & Hasan, 2000; Wu, 2010). Es por esto que BI es reconocido como un soporte importante

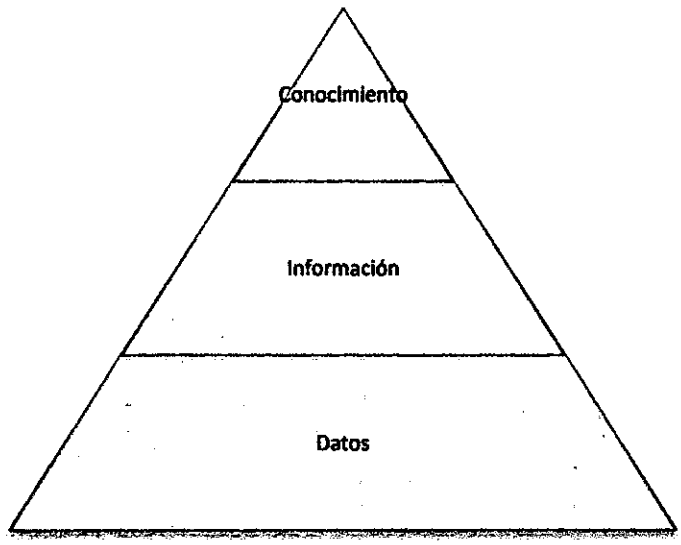
para el crecimiento de una compañía para tomar las mejores decisiones (Asghar, Fong, & Hussain, 2009).

5.2 KDD (Knowledge Discovery in Databases)

En las últimas décadas ha habido una recolección amplia de datos, esto debido al gran poder de procesamiento y almacenamiento a costos bajos de las máquinas computacionales.

No obstante dentro de esta aglomeración de datos yace una gran cantidad de información oculta, de gran importancia estratégica a la cual no se puede acceder por métodos estadísticos. Descubrir esta información es posible con la minería de datos, que aplicando varias técnicas entre las que está la inteligencia artificial se encuentran relaciones dentro de los mismos datos, permitiendo realizar representaciones y modelado de la realidad (información). Pero es el descubrimiento del conocimiento (KDD) quien se encarga de preparar los datos e interpretar los resultados obtenidos, los cuales dan significado a los patrones obtenidos (conocimiento). La relación entre datos, información y conocimiento se evidencia en la Gráfica 1 en donde el conocimiento se le es asignado un bajo volumen pero representa un alto valor.

Gráfica 1. Relación dato-información-conocimiento



Fuente: Mejia Juan, "Caracterización de algunas técnicas algoritmicas de la inteligencia artificial para el descubrimiento de asociaciones entre variables y su aplicación en un caso de investigación específic", 2009, pág 17

De este modo el valor real de los datos se encuentra en la información que se puede extraer de ellos, información que sirve para la toma de decisiones o la compresión de los fenómenos que se estudien. Esto se logra al implementar métodos analíticos avanzados a los datos de los negocios que permiten obtener la información adecuada para incrementar ganancias, reducir costos y mejorar la satisfacción del cliente.

Cabe destacar que la información que se ha generado y almacenado ha crecido considerablemente en los últimos años, se ha llegado a estimar que la cantidad de datos almacenados en el mundo se duplica cada 20 meses. Por consiguiente las organizaciones tienen gran cantidad de datos almacenados y organizados, pero no se pueden analizar en su totalidad.

Las sentencias SQL proceden a realizar un primer análisis, aproximadamente el 80% de la información se infiere a través de esta técnica. El 20% que es donde se encuentra la

información más relevante y en menos volumen (ver gráfica 1), requiere la utilización de técnicas más a la vanguardia.

La técnica del descubrimiento de conocimiento en las bases de datos (KDD) se enfoca en procesar grandes cantidades de datos para encontrar conocimiento útil, con base en ellos el usuario es capaz de usar la información obtenida para tomar acciones para su conveniencia propia o colectiva.

La información obtenida debe representarse visualmente o por lo menos de forma clara, esto con el objetivo primordial de que el usuario pueda interpretarla sin la utilización de conocimientos técnicos. El resultado no debe ser alterado por grandes volúmenes de datos. En este sentido el algoritmo elegido (algoritmo de minería de datos) para el descubrimiento de los datos debe ser lo más robusto posible.

Las metas del KDD son el de procesar la grandes cantidades de datos, identificar los patrones con mayor significancia y relevancia y posteriormente representarlos como conocimiento apropiado para satisfacer las metas del usuario. La primera se justifica con el hecho que entre más volúmenes de datos se tomen, el margen de certeza en la consistencia de los patrones que se obtengan van a ser mayor. La identificación de los patrones es el resultado del uso de algoritmos avanzados de minería de datos. La interpretación adecuada de los patrones es lo que se convierte finalmente en conocimiento.

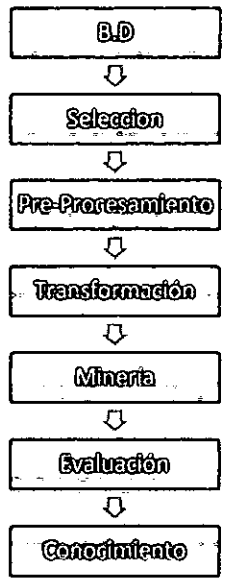
Con base en lo anterior la realización del proceso de obtención de conocimiento (KDD) consiste en una serie de pasos que van desde los métodos de minería de datos, específicamente los algoritmos a utilizar, para obtener los patrones que por último se interpretan para convertirse en conocimiento al cual se le puede sacar provecho en diferentes áreas, entre ella está los negocios y el comercio. De forma inversa la obtención de los patrones solo representa del 15% al 20% del esfuerzo total del proceso del KDD. El proceso de descubrimiento involucra varios pasos que son:

- Determinación de las fuentes de información: aquellas que pueden ser de utilidad y donde se consiguen.
- Diseño de un almacén de datos: que consigue unificar de forma operativa la información que se recoge.

- Seleccionar, limpiar y transformar los datos que se van a utilizar: se escogen los datos que se deben utilizar para el análisis. La limpieza y transformación de los datos se alcanza al implantar una estrategia para manejar valores incompletos, valores repetidos, y casos extremos en el peor de los casos.
- Seleccionar y aplicar el o los método de minería de datos apropiado: aquí se elige la selección de la tarea descubrimiento a realizar, por ejemplo, asociación, clasificación, regresión, etc.
- Evaluación, interpretación, transformación y representación de los patrones extraídos: se interpretan los datos y si es el caso se regresa a los pasos anteriores, eligiendo un nuevo algoritmo, otros datos, nuevas metas u otras estrategias. La interpretación hace uso del proceso de observación del sistema y de allí eliminar patrones redundantes o irrelevantes.
- Uso del conocimiento y su difusión: se incorpora el conocimiento adquirido al sistema con el fin de mejorarlo, esto normalmente incluye la resolución de conflictos potenciales. El conocimiento se obtiene con el fin de realizar acciones, ya sea para incorporarlos en el sistema o para almacenarlos y entregarlos a las personas interesadas. En este sentido el KDD implica un proceso interactivo e iterativo con el sistema en el que se involucra la utilización de algoritmos de minería de datos.

En la Gráfica 2 se ilustras los pasos de la técnica KDD:

Gráfica 2. Etapas del proceso de KDD



Fuente: Mejia Juan, "Caracterización de algunas técnicas algoritmicas de la inteligencia artificial para el descubrimiento de asociaciones entre variables y su aplicación en un caso de investigación específic", 2009, pág 17

5.3 Minería de datos o Data Mining

La minería de datos o data mining es un mecanismo para obtener información valiosa en grandes volúmenes de datos, es decir convertir los datos en conocimiento. Es una de las fases del KDD que representa el 20% del proceso. Utilizando algoritmos específicos la minería de datos permite descubrir, patrones, tendencias a través del análisis de los datos.

Con el uso de modelado se construye un modelo en una situación donde se conoce la respuesta y luego se aplica en otra situación donde se desconoce la respuesta. Los tipos de modelos son: predictivo y descriptivo:

- Modelo predictivo: este responde preguntas sobre datos futuros.
- Modelo descriptivo: proporciona información sobre la relación entre los datos.

La aplicación de la minería de datos en el mercadeo permite analizar base de datos de clientes o prospectos para obtener patrones de conductas o tendencias del mercado. También permite localizar problemas de atención al cliente, segmentación del mercado, perfiles de clientes, análisis de canasta, focalización de cliente y campañas promocionales, entre muchas otras.

La minería de datos nace para ayudar a comprender el contenido de un conjunto de datos. Para tal fin, hace uso de prácticas estadísticas, algoritmos de búsquedas cercanos a la inteligencia artificial, las bases de datos y el procesamiento masivo. Una definición común es: "la minería de datos es un paso en el proceso de KDD que consiste en aplicar análisis de datos y algoritmos de descubrimiento que producen una enumeración particular de patrones (o modelo) sobre los datos" (Fayyad, Piatetsky-Shapiro, & Smyth, 1996). Desde un punto de vista empresarial se define como: "La integración de un conjunto de áreas que tienen como propósito la identificación de un conocimiento obtenido a partir de las bases de datos que aporten un sesgo hacia la toma de decisión" (MOLINA & RIBEIRO, 2010).

El concepto de minería de datos no es nuevo. Ya desde los años 70 los estadísticos manejaban términos como data fishing, *data mining*, o *data archaeology*, con la idea de encontrar correlaciones sin una hipótesis previa en las bases de datos con ruido[1]. Llegado los años 80, Rakesh Agrawal, Gio Wiederhold, Robert Blum y Gregory Piatetsky-Shapiro, entre otros empezaron a arraigar los términos de data mining y KDD. Estas tecnologías han sido punto de encuentro entre personas pertenecientes al ámbito académico y de negocios.

Las técnicas de minería de datos son el resultado de largos procesos de investigación y desarrollo de productos a través de la historia. Esta evolución se inició cuando los datos de los negocios empezaron almacenarse en computadores, siguiendo las mejoras para acceder a los datos, hasta la posibilidad de los usuarios para navegar por los datos en tiempo real. Aprovechándose de este proceso de evolución la minería de datos lleva los datos más allá del acceso y navegación de los mismos hacia la entrega de la información

[1] En teoría de la información el ruido es el comportamiento poco preciso de los datos, por ejemplo una base de datos en el cual sus datos no representan de forma lineal el comportamiento del sistema que almacena

prospectiva y proactiva. La aplicación de minería de datos en los negocios esta respalda con el uso de tecnologías que están lo suficientemente maduras:

- Algoritmos de minería de datos
- La recolección masiva de datos en sistemas de bases de datos
- Computadores con potente procesamiento

También se caracteriza por explorar en profundidad las bases de datos, como son los almacenes de datos o lo que es lo mismo *data warehouse* que tienen información almacenada durante varios años. Las bases de datos pueden ser grandes en anchura (campos) como en profundidad (cantidad de registros). El analista debe limitar el número de campos a analizar, siendo estas las variables a analizar, a solo aquellas en la que están directamente relacionados con el comportamiento a evaluar. Por otro lado en mayor cantidad de registros produce menos errores de estimación y desvíos.

Dependiendo del caso los datos pueden estar acumulados en almacenes de datos y en otros casos en servidores de una intranet o en internet. Debido a la gran cantidad de datos muchas veces es necesario utilizar procesamiento en paralelo, con el fin de acelerar el proceso de extracción de información en cuestiones de minutos.

Las técnicas mayormente usadas para la extracción de información en la base de datos en la minería de datos son:

- Arboles de decisiones: son estructuras de forma de árbol que representan conjuntos de decisiones. Estas decisiones generan reglas para la clasificación de un conjunto de datos. Los Métodos específicos de árboles de decisión incluyen Arboles de Clasificación y Regresión (CART: *Classification And Regression Tree*) y Detección de Interacción Automática de Chi Cuadrado (CHAI: Chi *Square Automatic Interaction Detection*)
- Redes neuronales: son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en la que funciona el sistema nervioso de los animales.
- Algoritmos genéticos: técnicas de optimización que usan procesos tales como combinaciones genéticas, mutaciones y selección natural en un diseño basado en los conceptos de evolución.

- Regla de inducción: obtienes los resultados a través de condicionales lógicos (*if, else*)

Los tipos de información que extrae la minería de datos a través de las técnicas mencionadas son cinco:

- Asociaciones.
- Secuencias.
- Clasificaciones.
- Agrupamientos.
- Pronósticos.

5.4 Etapas Minería de datos

Aunque la minería de datos en distintos procesos puede proceder de diferentes formas, los procesos comunes en cualquier tipo de análisis se suele componer de cuatro etapas:

1. Determinación de los objetivos: Delimita los objetivos que el cliente desea bajo la supervisión de un especialista en data-mining.

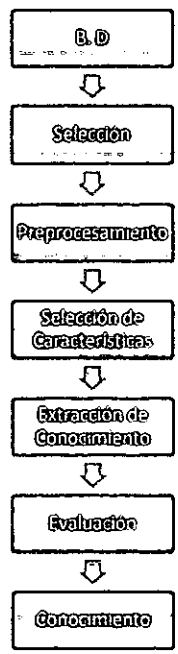
2. Pre procesamiento de los datos: hace referencia a la selección, limpieza, reducción y enriquecimiento de la base de datos. Este proceso toma el sesenta por ciento del tiempo que toma un proyecto de data-mining.

3. Determinación del modelo: en esta etapa se analizan los datos a través de métodos estadísticos. De acuerdo a los objetivos planteados y la tarea que se va llevar a cabo, se puede optar por utilizar algoritmos desarrollados en diferentes áreas de inteligencia artificial.

4. Análisis de los resultados: se verifican si los resultados son coherentes y se comparan con los análisis estadísticos. Finalmente el cliente fija si son novedosos y si le aportan un nuevo conocimiento que le consienta a considerar sus decisiones.

En la Gráfica 3 se ilustra estas etapas:

Gráfica 3. Etapas para obtención de conocimiento en base de datos



Fuente: Mejia Juan, "Caracterización de algunas técnicas algoritmicas de la inteligencia artificial para el descubrimiento de asociaciones entre variables y su aplicación en un caso de investigación específic", 2009, pág 17

Como se evidencia, el data-mining analiza datos, que sería la materia prima o bruta, utilizando algún algoritmo que vaya con de acuerdo al problema a estudiar. El resultado que se obtiene finalmente es información útil al estudio que se realice.

Un caso en donde este conocimiento fue obtenido y aplicado para el beneficio económico de la empresa es la cadena de supermercados WAL-MART, allí se descubrió que los compradores hombres impulsados por sus esposas a adquirir pañales, también llevaban cerveza en su gran mayoría. Por lo anterior, la cadena de supermercados decidió colocar los pañales junto a las cervezas para impulsar la venta de cervezas. El análisis concluye que se presentaba una asociación directa entre pañales y cervezas (Liu & Guan, 2009). Por tal razón se decidió colocar los pañales y las cervezas en el mismo

estante con el fin de inducir a aquellos hombres que solo compraban pañales a llevar también cerveza. Es de destacar que las inducciones se obtuvieron con el uso de algún algoritmo de asociación sobre la gran cantidad de datos que el supermercado almacenó.

5.5 Reglas de Asociación

Con base en el caso anterior se puede inferir que para realizar el análisis en las ventas de productos y poder encontrar relaciones de preferencia de compra entre los mismos se utiliza los algoritmos de inteligencia artificial, puesto que son los que se utilizan para analizar cómo se relacionan las variables para determinar un comportamiento específico y frecuente en un contexto. Las asociaciones son utilizadas a menudo para descubrir relaciones o correlaciones entre los conjuntos de ítems que llevan los consumidores en el mercado (Hornick, Marcadé, & Venkayala, 2007, pág 93). Por ejemplo nueve de cada diez personas que compran leche compran pan en la misma transacción.

Las reglas de asociación lidian son ítems discretos, formado por dos conjuntos de ítems (a partir de ahora llamado *Itemset*). Un *Itemset* llamado antecedente implica otro llamado consecuente. En el ejemplo anterior el antecedente es la leche y el consecuente corresponde al pan y se puede representar de la forma A → B (Leche → Pan).

La medida de calidad de la regla esta cuantificada por el soporte y la confianza. El soporte cuantifica que tan frecuente los ítems asociados a la regla ocurren juntos. La confianza indica la probabilidad de encontrar tanto el antecedente y el consecuente en la misma transacción, dada la frecuencia del antecedente (su soporte). Por ejemplo, de la Tabla 1, la regla Leche → Pan (Leche implica Pan) presenta sus valores de soporte y confianza como sigue.

Leche → Pan:

$$\text{Soporte} = 3 / 10 = 30\% \text{ y Confianza} = 3 / 4 = 75\%$$

Tabla 1: Representación de ventas de productos en una transacción

ID Transacción	Artículos Comprados
100	Leche, pan
200	Pan, leche, huevos
300	Huevos, cereal
400	Leche, pan, carne
500	Carne, cereal
600	Mantequilla, cereal
700	Leche, huevos
800	Cereal, huevos, mantequilla
900	Huevos, mantequilla
1000	Huevos, mantequilla

Fuente: Naranjo Roberto & Sierra Luz, "Software tool for analysing the family shopping basket without candidate generation", 2009, pag

Esto significa que hay un 75% de probabilidad de aquellas personas que compran leche lleven pan. En cambio la regla Pan → Leche obtiene un valor de confianza de 100%, pues en todas las transacciones donde aparece el pan está presente el artículo leche.

Los datos de entrada para obtener las reglas de asociación son comúnmente representados por una matriz bidimensional en la que cada fila representa una transacción y cada ítem corresponde a una columna. La literatura menciona cuatro formas de representar la matriz bidimensional (Shenoy et al., 2000):

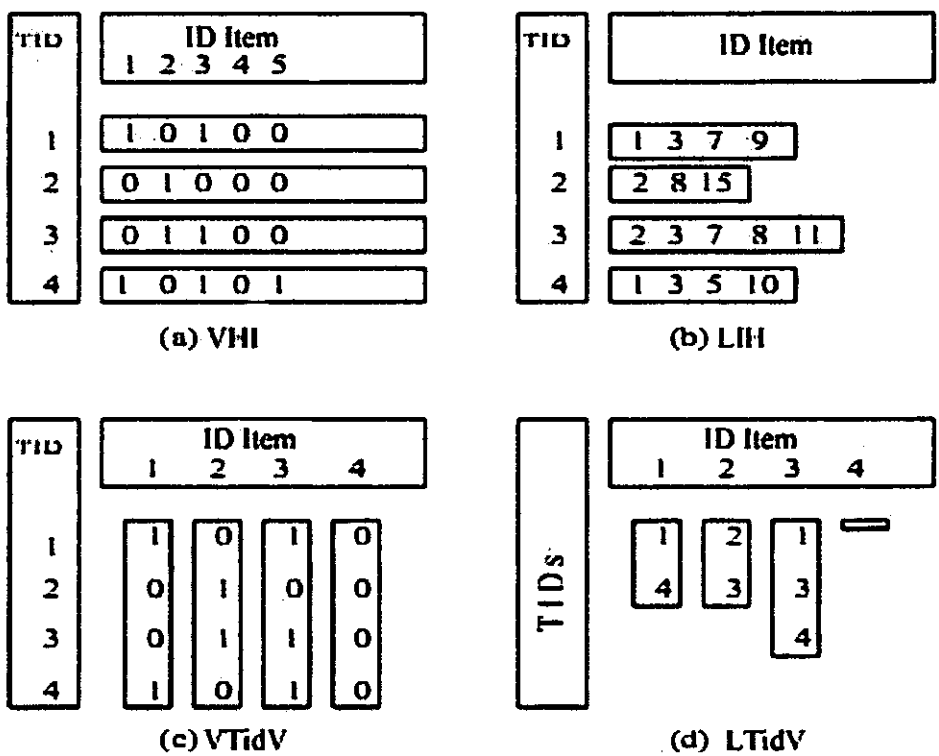
Lista de Ítems Horizontales (LIH): representa los artículos en una lista por cada transacción (Gráfica 4b).

Vector Horizontal de Ítems (VHI): por cada transacción se almacena un vector binario en donde el numero uno significa la presencia de un artículo y un cero la ausencia (Gráfica 4a).

Vector de Identificador de Transacciones Vertical (VTidV): La columnas de la matriz representa un artículo y almacena un vector binario que indica la presencia de dicho artículo en una transacción (Gráfica 4d).

Lista de Identificador de Transacciones Vertical (LTidV): al igual que la anterior las columnas representan artículos pero estas almacenan en una lista los identificadores solo de las transacciones en la que está presente el artículo en cuestión (Gráfica 4c).

Gráfica 4. Diferentes representaciones de la Matriz bidimensional



Fuente: Shenoy et al. "Turbo-charging Vertical Mining of Large Databases", 2000

La representación en lista tanto en vertical como horizontal son las más utilizadas por su bajo consumo de memoria. Las de vectores tiene la desventaja de representar explícitamente la ausencia de un artículo. A pesar que la representación en horizontal es más utilizada que la vertical esta última hace más fácil el cálculo de soporte interceptando los vectores binarios o las listas de los artículos.

La forma de recorrer los ítems para encontrar conjunto frecuentes también posee diferentes formas de realizarlo. El mecanismo está definido a la dirección del recorrido.

Si el recorrido se realiza desde los 1-itemset hasta el máximo posible se le denomina recorrido descendente. En contraste si el recorrido se realiza de forma inversa al descrito anteriormente (se inicia con un súper-conjunto de ítems hasta sus sub-conjuntos) se denomina ascendente. Junto a los recorridos, la heurística para generar el conjunto de ítems se da de dos formas: en amplitud donde primero se genera los (k-1)-itemsets antes de generar los k-itemsets aplicado en el trabajo "Fast Algorithms for Mining Association Rules" y "Multipass Algorithms for Mining Association Rules in Text Databases" (Agrawal & Srikant, 1994; Holt & Chung, 2001) y en profundidad implementados en "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach" y "Mining Frequent Itemsets using Patricia Tries" (Han, Pei, Yin, & Mao, 2004; Pietracaprina & Zandolin, 2003) que toma cada ítem y obtiene todos los súper-conjuntos que se pueden generar una vez no se puede tomar más ítems se toma el siguiente ítem para realizar el mismo proceso.

5.6 Algoritmos

A continuación se presentan algunos de los algoritmos de asociación más citados, cada uno utilizan diferentes estrategias de recorrido y representación de la matriz de datos;

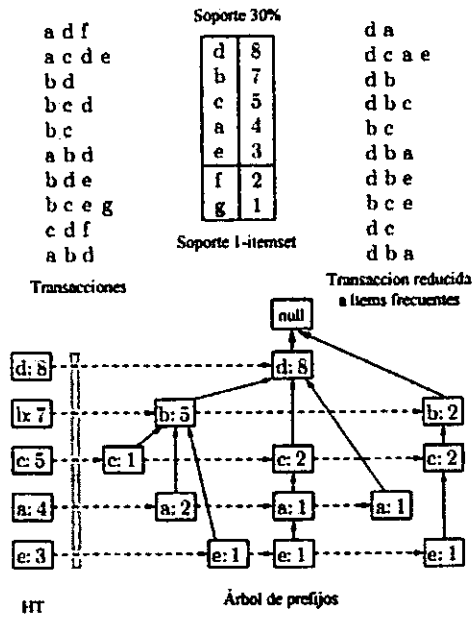
Apriori: Este algoritmo y sus variaciones usan recorrido en amplitud, y almacena utilizando LIH, es presentado por Agrawal en "Fast Algorithms for Mining Association Rules" (Agrawal & Srikant, 1994) en el que se recorre cada transacción t para generar los candidatos C_K que posteriormente extrae los ítems frecuentes L_K. Busca primero todos los conjuntos frecuentes unitarios (contando sus ocurrencias directamente en la base de datos) conocido como 1-*itemsets*, se mezclan estos para formar los conjuntos disjuntos de ítems candidatos de dos elementos (2-*itemsets*) y seleccionan entre ellos los frecuentes (L_2). Considerando la propiedad de los conjuntos de ítems frecuentes, se vuelve a mezclar estos últimos que se denotan $L_{(k-1)}$ para generar un nuevo conjuntos disjuntos de k-*ítemsets* candidatos (C_k) de allí se obtienen los frecuentes L_k. Sucesivamente se repite el proceso hasta que en una iteración no se obtengan conjuntos frecuentes L_K (Acuna, 2010; Agrawal & Srikant, 1994; Shi & Zhao Yu-lin, 2009; Xindo & Vipin, 2009).

***AprioriTid*:** esta variación de Apriori hace un recorrido a la base de datos *D* para generar los 1-itemsets, posterior a ello crea otra estructura en la que se almacena en pares ordenados <Tid, X_k>, donde Tid es el id de la transacción y X_k corresponde a los ítems candidatos frecuentes de k elementos que se encuentran en la transacción, aquellos Tid que no tengan ítems candidatos no son incluidos en la estructura. En cada iteración k se recorre los registros que se generan en la iteración k-1. Cuando k es grande los registros generados son menores que las transacción pero cuando son pequeños los registros son más grandes que la transacción inicial.

Apriori Híbrido: toma la ventaja de los algoritmos anteriores, recorre las transacciones generando ítems frecuentes, a partir de cierto k-itemset, usa la estrategia propuesta en el *AprioriTid*. Así se evita crear registros muy grandes en los primeros *k-itemsets*.

***FP-growth (frequent pattern growth)*:** A diferencia del algoritmo Apriori su estrategia de recorrido es en profundidad y ha sido implementado en (Borgelt, 2005; Liu & Guan, 2009). Usa una estructura de árbol llamada FP-Tree (árbol de prefijos) (Grahne & Zhu, 2003), en el que cada nodo es etiquetado con un ítem y su soporte y enlazados descendentemente. Para formar la estructura primero se extraen los 1-temsets frecuentes, luego se ordenan por el soporte de mayor a menor otra estructura llamada HT (Header Table) con la cual se construye el árbol de prefijos. La Gráfica 5 muestra como se construye el árbol de prefijos a partir de diez transacciones y un soporte mínimo de 3 elementos. Construidas estas estructuras se generan los ítems frecuente tomando cada ítem de forma ascendente a su soporte. Se extrae el árbol en la que las ramas se encuentra el ítem, luego se procesa nuevamente este árbol siguiendo los pasos anteriores pero sin incluir los nodos hojas. Esta estructura muestra eficiencia siempre y cuando el conjunto de datos no sea muy disperso, en este caso la construcción de la estructura y su recorrido se hacen demasiado costosos.

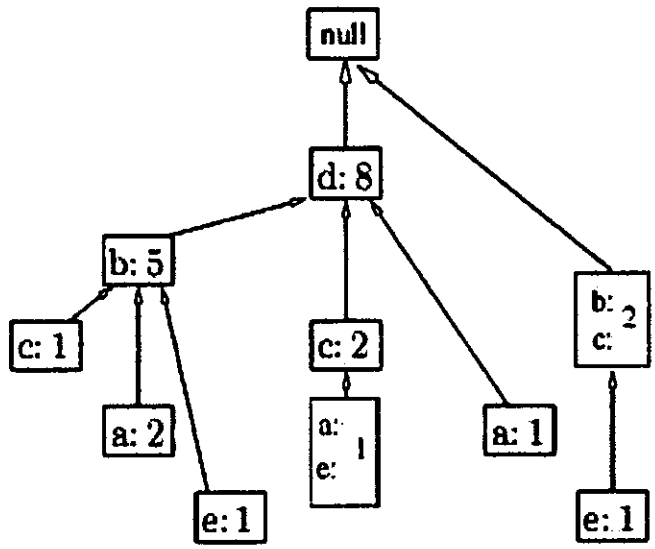
Gráfica 5. Construcción del árbol de prefijos.



Fuente: Borgelt Christian, "An Implementation of the FP-growth Algorithm", 2005

Patricia Mine: compacta la estructura utilizada en FP-growth colocando en un solo nodo todos ítems que son consecutivos y tienen el mismo soporte. Esta estructura se le conoce como Patricia Trie (Pietracaprina & Zandolin, 2003), la Gráfica 6 muestra la representación de en Patricia Trie del árbol de la Gráfica 5. El método de obtener los ítems frecuencia es similar al del algoritmo anterior la única variante es la estructura del árbol. A pesar de compactar la estructura Patricia Mine presenta las mismas deficiencias de FP-growth en datos dispersos.

Gráfica 6. Representación de la estructura Patricia Trie



Fuente: Pietracaprina Andrea, "Mining Frequent Itemsets using Patricia Tries", 2003

5.7 Referentes de proyectos locales

Los algoritmos mencionados han sido probados en los trabajos referenciados de "Fast Algorithms for Mining Association Rules" (Agrawal & Srikant, 1994), "Mining Association Rules between Sets of Items in Large Databases" (Agrawal et al., 1993), "A fast APRIORI implementation" (Bodon, 2003), "Applying Business Intelligence in Marketing Campaign Automation" (Gaik Yee et al., 2000), "Mining frequent patterns without candidate generation" (Han et al., 2000), "Application in Market Basket Analysis Based on FP-growth Algorithm" (Liu & Guan, 2009) y "Mining Frequent Itemsets using Patricia Tries" (Pietracaprina & Zandolin, 2003). Cada uno hace su implementación para obtener ítems frecuentes y evalúan su rendimiento con diferentes valores de soporte. Igual sucede con el trabajo realizado en la universidad de Nariño (Timarán Pereira, Andrés, Ramírez, Alvarado, & Guevara, s.d.) en la que se compara el rendimiento de tres algoritmos de asociación. Sin embargo ninguno menciona como hallar las reglas entre los ítems frecuentes y por ese motivo no realizan comparaciones

al variar la confianza. Tampoco hacen referencia a implementarlo en un software que almacene las ventas de productos.

Dentro investigaciones de ámbito local el trabajo "Modelo de decisión para el proceso de mercadeo de nuevos vehículos en GM Colmotores" (Gómez Vargas & Castillo, 2005) hace un análisis de la posición de la empresa en el mercado, se verifica el proceso de toma de decisiones relacionado con los diferentes productos, se estudia los problemas del entorno de la empresa y se concluye con la creación de un modelo de decisión, para esto se utilizo un software llamado HUGIS que sirvió de motor para la generación de redes bayesianas relacionales obtenidos de los datos procesados, el análisis ayudado por los investigadores les favoreció a identificar las relación "causa-efecto" de la salida de un nuevo producto al mercado; sin embargo este sistema no guarda la relación de los productos entre ellos, se podría utilizar como punto de comparación para identificar cuáles son las características que tiene un artículo sobre otro de ser vendido. También es factible aclarar que solo se desarrollo un modelo haciendo uso del software HUGIS y no se realizó uno a la medida.

En la universidad del atlántico en el trabajo "Inteligencia artificial en la gestión financiera empresarial" (Sosa Sierra, 2004) se hace mención descriptiva a varias técnicas de la inteligencia artificial, pero no se hace uso ni se desarrolla una aplicación dentro de la investigación dada, como son redes neuronales, algoritmo de lógica difusa, algoritmo genético, la teoría de rough sets y como se pueden aplicar en un ámbito de gestión, planificación, ejecución y control de la administración financiera para apoyar las decisiones, dentro de esta investigación se muestran como cada sistema de la inteligencia artificial puede ser solución a un problema de ámbito financiero: sistemas de predicción por medio de la lógica difusa, capacidad financiera de una empresa para absorber un préstamo gracias al algoritmo genético, estudios de evaluación del comportamiento de las acciones de las empresas en el mercado de valores por medio de redes neuronales, descubrimientos de tendencias en el negocio (un ejemplo puede ser las tendencias de compras de los consumidores) haciendo uso de técnicas de minería de datos, entre otros. También se menciona que es necesario seguir investigando bajo qué condiciones estas herramientas pueden ser una solución más eficiente para desarrollar modelos y/o herramientas que ayuden a la toma de decisiones financieras más acertadas.

Otra investigación importante en el tema es el trabajo "Software tool for analysing the family shopping basket without candidate generation" (NARANJO & Sierra, 2009) realizado en la Universidad Nacional de Colombia, en el cual se hace un estudio de los datos para determinar cuál es la probabilidad de que un producto pueda ser comprado por medio de la relación encontrada por otros productos. Dentro de la investigación se mencionaron las reglas de asociación y se implementó el algoritmo FP-Growth, por su aplicación y solución de los datos de manera de árbol de decisión. Las pruebas a la herramienta desarrollada las realizaron con una base de datos de la compañía FoodMart (tienda de cadena dispersa en Estados Unidos, Canadá y México) con un soporte de 20% y confianza de 40%. Con la herramienta desarrollada se descubrieron 52 reglas mientras que herramientas como Weka se descubrieron 64. Es evidente que el prototipo no obtuvo todas las que se podían extraer, además solo 49 concordaban con los resultados de Weka. En conclusión en el trabajo se desarrolló el algoritmo de FP-Growth con una efectividad de 76% para encontrar reglas de asociación. Tampoco se menciona que algoritmo se utilizó con Weka para obtenerlas y no se varían los valores de soporte y confianza para ver el comportamiento de la herramienta desarrollada.

Así se puede también mencionar el trabajo de investigación "Aproximación al proceso de toma de decisiones en la empresa barranquillera" (Cabeza de Vergara & Muñoz Santiago, 2004), el cual hace un estudio descriptivo del proceso de cómo se maneja la toma de decisiones en las empresas y sus herramientas más relevantes, aquí se mencionan ciertas características como conclusiones a 77 empresas visitadas en la ciudad de Barranquilla. De los resultados de las encuestas se obtuvo que el 70% de la muestra dedica el 20% de su tiempo a decisiones en el mercadeo de productos y solo el 21% se apoya en programas computacionales. Los autores ven preocupante el poco uso de herramientas computacionales puesto que con ellos es más rápido analizar grandes cantidades de información y se compensaría con el poco tiempo invertido a decidir. También 24 de las 77 empresas alegaron que se les hacía necesario un modelo para analizar las tendencias de compras de los consumidores. Hay que anotar que este trabajo solo menciona los modelos computacionales que se podrían usar en las áreas producción, recursos humanos, mercadeo, entre otros, pero no profundiza que técnicas de modelado serían útiles, ni que herramientas existen a la fecha. Se pudo haber hecho profundidad en técnicas de inteligencia artificial como en el trabajo "Inteligencia

artificial en la gestión financiera empresarial" (Sosa Sierra, 2004) y mencionar las herramientas informáticas con las que se basaban algunas empresas para sus decisiones.

En el trabajo "Análisis de desempeño de EquipAsso: Un algoritmo para el cálculo deItemsets frecuentes basado en operadores algebraicos relacionales" (Timarán, Calderón, Ramírez, Alvarado, & Guevara, s.d.) Elaborado en la Universidad de Nariño se realiza una comparación del rendimiento computacional del algoritmo EquipAsso con los algoritmos Apriori y FP-Growth. EquipAsso, explican los autores, obtiene los ítems frecuentes utilizando operadores algebraicos relacionales Associator (α) y EquiKeep (χ). Associator se encarga de generar todos los posibles Itemsets de una tupla, mientras EquiKeep restringe el número de ítems que pueden estar en un Itemset. Los registros en la base de datos deben estar representados en forma de vector binario para que el algoritmo pueda funcionar. El algoritmo EquipAsso se realiza utilizando consultas SQL utilizando los operadores α y χ en una base de datos Posgret. Por su parte Apriori y FP-Growth se implementan utilizando Tariy (programa de minería de datos no acoplada con un motor de base de datos). Los resultados arrojan que EquipAsso es más eficiente que Apriori para soporte bajos; pero para soporte altos, el rendimiento es similar a Fp-Growth y Apriori. Debido a que EquipAsso estuvo acoplado a la base de datos, los resultados son discutibles teniendo en cuenta que los otros dos algoritmos no lo estaban. El proyecto no realiza una interfaz para el algoritmo, pero recomiendan que posteriormente se realice. Otro aspecto negativo se aprecia que no se menciona si es posible generar las reglas de asociación utilizando los operadores relacionales. Como se mencionó antes los datos de las transacciones se representaron en vector binario, lo que significa que se desperdicia memoria guardando los datos que no hacen parte de una transacción.

Como se puede observar la mayoría de las investigaciones a nivel local se centran en realizar estudios descriptivo de cómo las técnicas de inteligencia artificial (IA) pueden ayudar a comprender mejor el funcionamiento de un negocio para tomar decisiones en un área como puede ser el de mercadeo. En otras solo implementan los algoritmos para probar su rendimiento, pero no lo integran a un software concreto, sin embargo en la mayoría invitan a seguir investigando con la finalidad de encontrar otros campos en los

que pueda ser útil la minería de datos y realizar desarrollo de herramientas usando estas técnicas.

Al momento de aplicar una investigación para la aplicación de las técnicas de minado de datos y extracción de reglas de decisión se debería tener en cuenta el costo del software y la capacidad adquisitiva del uso de las herramientas. Por esta razón se hace necesario un software ligero que contenga solo funcionalidades que se necesitan en una microempresa, obtención de las tendencias de compras y almacenaje de los inventarios y las ventas. Con el fin de que el sistema software contará con la ventaja de poder adaptarse a las bases de datos más usadas tanto libres como pagas, una disponibilidad en línea a través de tecnologías web y funcionalidades de análisis de las ventas durante un periodo determinado de la empresa beneficiaria, escogiendo rangos semanales, mensuales e ir extendiendo el análisis a procesos semestrales y anuales de .forma automática. De esta manera se podrá utilizar para cualquier empresa del mercado de ventas buscando una mejora de las ventas.

5.8 Herramientas

Herramientas de minería de datos como *WEKA* («Weka 3 - Data Mining with Open Source Machine Learning Software in Java», s.d.), *RapidMiner* («Rapid - I», s.d.) y *KNIME* («KNIME | Konstanz Information Miner», s.d.) tienen entre sus funciones crear reglas de asociación a partir de un dataset permitiendo realizar tareas de *Business Intelligent*. También realizan el minado utilizando módulos para base de datos, sin embargo los datos deben estar des-normalizados y en forma de vector binario para uso de algoritmos de asociación. Esto implica que la migración de los datos normalizados a su forma de vector binario consuma tiempo en el proceso y se cree información duplicada en la base de datos. Por estas razones integrarlas a una aplicación que maneja las ventas de los productos sería costosa en procesos. La aplicación propuesta no usará vector binario e implementará una modificación del algoritmo Apriori con el cual se esperaría un mayor rendimiento al tradicional.

WEKA: es una herramienta desarrollado por la universidad de *Waikato en el año 1993*, para el aprendizaje automático y minería de datos. Es de libre distribución bajo la

licencia *GNU GPL* y no está disponible en español. Tiene integrado el algoritmo Apriori para obtener ítems frecuentes y reglas de asociación, sin embargo los datos deben estar agrupados en forma de vector binario en un archivo plano o en la base de datos, cuyo problema radica en que se declaran explícitamente los ítems que no están en una transacción.

Knime: es una plataforma de análisis de datos basado en la plataforma eclipse, fue desarrollado por la Universidad de *Konstanz*, Alemania. La compañía ofrece contratos de consultoría, formación y soporte técnico. Al igual que WEKA es multiplataforma y se distribuye bajo la licencia *GPL*. Su fuerte es representar el análisis lo más intuitivamente para el usuario utilizando diferentes tipo de diagramas (principalmente de barra y de circular). Esta estrategia es válida para aplicar al software a desarrollar puesto que oculta la parte técnica y muestra solo la información útil que un usuario le interesa.

RapidMiner: es otro programa para el análisis y minería de datos disponible en versión empresarial (Enterprise edition) y de comunidad (community edition) en idioma español y alemán. Está escrito en java y permite integrar los algoritmos de *WEKA*. Puede utilizarse a través de la línea de comando o por la interfaz gráfica. RapidMiner realiza el minado directamente en la base de datos, en lo que tiene que ver con algoritmos de asociación, cuyos datos deben estar representados en forma de vector binario al igual que WEKA.

5.9 Mercadeo y ventas

El mercadeo es entendido como los mecanismos para satisfacer las necesidades de los clientes. Está directamente relacionado con realizar ventas de los productos de forma óptima, lo que implica que un producto sea ofrecido para que un consumidor se interese en él. Un buen trabajo de mercadeo implica que al identificar las necesidades de los clientes, los productos con un valor superior, y distribuirlos eficientemente, entonces dichos productos serán vendidos con mayor facilidad (Kotler, Armstrong, Saunders, & Wong, 1999, pág 9).

El uso de nuevas tecnologías, como lo diría Kotler, no se limita a colocar una empresa en el internet o implementar un CRM o un ERP. Destaca que el uso de alta tecnología debe ir encaminado a realizar análisis del comportamiento de los clientes y de mantener al vendedor siempre informado del estado del inventario y el comportamiento de los productos. También aclara que el mercadeo se ha convertido en una contienda en la que se encuentra por encima la información sobre el poder de las ventas.

Son variadas las herramientas utilizadas actualmente para la promoción y venta de artículos dentro del marco empresarial, muchos de esos métodos son la venta directa de artículos que se realizan puerta a puerta, la venta por catálogos, revistas, correo, por teléfono, promoción y por medios de información.

Esto ha provocado una demanda en el crecimiento de los métodos donde el usuario puede ver el producto y acceder a él rápidamente. Las empresas dentro de su estructura disponen de muchas herramientas para sacar provecho a las posibilidades de comunicación entre sus proveedores y los clientes de esta manera logran fortalecer su mercado.

Para una empresa es indispensable tener sistemas colaborativos que le ayuden a fortalecer sus áreas de mayor crecimiento y expansión como es el área de mercadeo y ventas, este le ayudara a identificar los productos más relevantes y estrategias para expansión del mercado, y atracción de nuevos clientes.

Para ayudar a generar las estratégicas que ayuden a la tarea del área de ventas y mercadeo se han dispuesto de los medios electrónicos los cuales son las llamadas nueves formulas comerciales, de ahí que se toda la información se disponga dentro de la web, y otros medios similares. Al ser dispuestos en estos medios pueden ser analizarlos en un menor tiempo dando respuesta a nuevas tendencias. Con esto se propuso adaptar este modelo a un sistema web de análisis que encuentre posibles tendencias de los productos, y características externas que ayuden a generar estrategias para el área en cuestión.

5.10 Sistemas Gestores de Base de datos

La base de datos, se comprende como una reunión de datos que están relacionadas con un contexto, almacenados de forma automática para su posterior consulta. Los Sistemas de Gestión de Base de datos, muchas veces abreviado SGBD, es el conjunto de programas que acceden a la base de datos. El fin primordial de un SGBD es obtener e ingresar información de una base de datos de una forma práctica y eficiente.

Grandes cantidades de información son manipuladas por diseños óptimos de sistemas de bases de datos. De allí los sistemas de bases de datos deben brindar la fiabilidad de la información almacenada. Mecanismos como la definición de la estructura de cómo se guardaría la información y su manipulación son aspectos que conciernen a la gestión de las bases de datos.

Los sistemas de bases de datos son usadas en la mayoría de negocios y proveedores de servicios formando una parte esencial en casis todas las empresas actuales: bancos, universidades, líneas aéreas, finanzas, ventas, recursos humanos, línea de producción, entre otros. Su uso se expandió en las empresas a finales del siglo XX. Las personas del común interactuaban con las bases de datos inconscientemente cada vez que hacían uso de un cajero automático y las reservas por teléfono de las líneas aéreas.

El acceso directo a las bases de datos fue producido por la masificación del internet en los años 90[26]. Muchas compañías cambiaron sus sistemas telefónicos a la interfaz web y agregaron más servicios. De allí cuando una persona consulta su cuenta a través de internet, en realidad estaban consultando la base de datos de la entidad bancaria, también cuando se reserva libros a una biblioteca por una la web. La efectividad en el almacenamiento y consulta de datos en una base de datos es consecuencia de un buen diseño de la misma. La meta del diseño de base de datos relacional es la generación de de relaciones entre las tablas que permita ingresar la información sin redundancia y a la vez recuperar de forma fácil la información. Existen reglas que garantizan lo anterior conocidas como las formas normales. Existen 3 formas normales básicas que garantizan la no redundancia de información:

La primera forma normal se cumple si cada uno de los elementos de una tupla es indivisible, es decir, impide que un atributo de una tupla pueda tomar múltiples valores. Así el atributo nombre de una tabla usuarios no pertenece a la primera forma normal, puesto que nombre puede dividirse en nombres y apellidos.

Si se cumple la primera forma normal, la base de datos estará en la segunda forma normal si todos los elementos de una tupla dependen funcionalmente de la llave primaria. Todos los atributos que dependen parcialmente de la llave primaria deben formar otra tabla con esos atributos y con llave foránea de la tabla de la que se desprenden.

Para que una tabla este en tercera forma normal, debe estar en segunda forma normal segunda forma normal, y se asegura que cada elemento de la tupla puede acceder transitivamente a datos relacionales de otra tupla. Una no está en tercera forma normal cuando uno de los atributos depende funcionalmente de otro que no sea clave. Para solventar el problema, los atributos don aislados en otra tabla que apunta con una llave foránea a la tabla que se desprende.

La recuperación y almacenamiento de la información es tan importante como la estructura, y las bases de datos relacionales poseen mecanismos para ellos. El lenguaje SQL (*structured query language*), es un leguaje que permite realizar diferentes tipos consultas y operaciones sobre una base de datos. La información de interés es posible obtenerla con el manejo del algebra relacional de este lenguaje. Además soporta características para de definición de datos, modificación y restricciones de seguridad.

El desarrollo de SQL surge por IBM en el año de 1970, como un proyecto para implementar el álgebra relacional en bases de datos relacionales. Recibió el nombre de SEQUEL (*Structured English QUEry Language*). El sistema de base de datos relaciones *System R* desarrollado también por IBM en 1977 hizo uso amplio de este lenguaje. Más tarde en 1979 ORACLE lo introdujo en un programa comercial. Su avance y desarrollo lo llevo a convertirse en lo que ahora se conoce como SQL (*Structured Query Language*, Lenguaje estructurado de consultas). SQL es implementado como el patrón de referencia de las bases de datos relaciones.

En año 1986 la ANSI (*American National Standards Institute*, Instituto Nacional Americano de Normalización) y la ISO (*International Standards Organization*, Organización Internacional de Normalización), aprueban la primera versión estándar del lenguajes llamado SQL-86. Para cubrir necesidades de los desarrolladores que no son abarcados en la primera versión es lanzado en 1992 una revisión ampliada y mejorada del estándar, SQL-92. La más reciente es la norma SQL-1999 que añade nuevas cualidades al lenguaje de SQL-92. Sin embargo algunos sistemas de base de datos no alcanzan a soportan todas las funcionalidades de SQL-92 y otros añaden características no estándares.

Las tareas que permite SQL sobre la base de datos se clasifican en dos principales. El Lenguaje de Definición de Datos (*Data Definition Language*, DDL por sus siglas en inglés) y DML (Data *Manipulation Language, Lenguaje de Manipulación de datos*). La modificación de la estructura de la entidades de la base de datos es realizado por El Lenguaje de Definición de Datos, sus realizaciones básicas son ALTER, CREATE, DROP y TRUNCATE. Para realizar la consulta y modificación de los registros es utilizado el Lenguaje de Manipulación de datos, cuyas operaciones básicas las realiza las sentencias DELETE, INSERT, UPDATE.

Tabla 2: Sentencias DLL

Comando	Descripción	Ejemplo
ALTER	Modifica la estructura de un objeto. Agregar o quita campos de una tabla, modifica el tipo de un campo, añade o elimina índices de una tabla, cambia el nombre de una tabla, etc.	ALTER TABLE estudiante RENAME alumno;
CREATE	Construye un objeto en la base de datos: tabla, índice, *trigger*, procedimientos, etc.	CREATE TABLE estudiante (nombre char(50), apellido char(50))
DROP	Elimina objetos de una base de datos: tabla, índice, trigger, procedimientos, etc.	DROP TABLE estudiante
TRUNCATE	Elimina todos los registros de una tabla. Es más rápido que la sentencia DELETE de DML porque borra la tabla y la reconstruye sin hacer en ningún momento una transacción.	TRUNCATE TABLE estudiante

Tabla 3: Sentencias DML

Comando	Descripción	Ejemplo
DELETE	Elimina uno o más registros de una tabla	DELETE FROM estudiante WHERE id = 2
INSERT	Agrega uno o más registros a una tabla	INSERT INTO estudiante VALUES ('Rafael' ,Fernández');
UPDATE	Modifica los valores de un conjunto de datos existentes en una tabla.	UPDATE estudiante SET nombre = "Juan" ,apellido = "Perez" WHERE nombre = "Rafael"

La obtención de información en SQL se basa en la teoría del álgebra relacional, las estructura para obtener información se fijan criterios. Los comandos básicos para tal fin corresponden a tres cláusulas: SELECT, FROM Y WHERE.

5.10.1 Cláusulas SELECT y FROM

La proyección en el álgebra relacional corresponde con la cláusula SELECT. El resultado de la consulta es una relación que devuelve todos los atributos seleccionados. Los atributos que se seleccionan provienen de una tabla especificada por la cláusula FROM. La estructura es la siguiente:

SELECT "nombre_columna" FROM "nombre_tabla"

5.10.2 Cláusula WHERE

Cuando se desea seleccionar condicionadamente los datos de una tabla la clausula WHERE cumple esa función. También se puede unir predicados con los conectores lógicos AND, OR y NOT en lugar de los símbolos matemáticos ∧∨□□. Los operadores de los conectores pueden ser expresado con los signos matemático <, <=, >, >=, = y <>.

Estructura en una selección:

SELECT "nombre_columna" FROM "nombre_tabla" WHERE "condición"

WHERE es útil cuando se desea obtener un grupo de registros, por ejemplo si se desea obtener los registros de todos los estudiantes hombres mayores de edad se usaría una sentencia como la sentencias:

*SELECT * FROM estudiante WHERE edad >=18 AND genero = "M"*

Para realizar consultas complejas, como agrupación, ordenación, se hace necesario el uso de las funciones de agregado, comparación y operadores lógicos. Las funciones de agregado se usan dentro de la cláusula SELECT en un grupo de registros para obtener un valor único: un conteo de registro, el promedio de un campo, el mínimo valor de un campo, entre otras. Por ejemplo para obtener el promedio de edad de alumnos en una tabla que almacena los estudiantes de un colegio, se usaría la función AVR (*average*, promedio en inglés):

SELECT AVG (edad) FROM estudiante

Tabla 4: Funciones de Agregado

Función de Agregado	*Función*
AVG	*Calcula el promedio de los valores de un campo seleccionado.*
COUNT	*Devuelve la cantidad de registros seleccionados*
SUM	*Se utiliza para devolver la suma de todos los valores de un campo seleccionado.*
MAX	*Devuelve el valor más alto de un campo seleccionado. Si todos los valores son iguales toma este como el máximo a entregar.*
MIN	*Devuelve el valor más bajo de un campo seleccionado. Si todos los valores son iguales toma este como el mínimo.*

Además de la consultas para obtener información de una tabla, SQL posee mecanismos para reunir relaciones. Los campos que relacionan a las tablas se especifican en la cláusula WHERE; mientras que las tablas involucradas se mencionan en la cláusula

FROM. En las tablas 4 y 5 se las relaciones estudiante y cursos, cada esudiante pertenece a un curso identificado por el atributo "id_curso", sin embargo hay un estudiante que pertenece a un curso que no existe (Samanta), y por otro no todos los cursos pertenecen a por lo menos un estudiante (Física y literatura). La diferencia entre reuniones de tablas, conocidas como *joins* en inglés, se puede representar con algunos ejemplos sobre estas tablas.

Tabla 5: Tabla estudiante

nombre	Id_curso
Juan	51
Madison	51
Gaspar	54
Samantha	56

Tabla 6: Tabla curso

id	Nombre
51	Matemáticas
52	Física
54	Biología
55	Literatura

La combinación básica INNER JOIN, cruza las dos tablas y devuelve una relación en donde solo permanecen los atributos que se relacionan entre dichas tablas. Ejemplos:

*SELECT * FROM estudiante INNER JOIN curso ON*

estudiante.id_curso = curso.id

Tabla 7: INNER JOIN estudiante y curso

estudiante.nombre	estudiante.id_curso	curso.id	curso.nombre
Juan	51	51	Matemáticas
Madison	51	51	Matemáticas
Gaspar	54	54	Biología

Para obtener todos los registros que se encuentren en ambas tablas sin importar si poseen datos equivalentes se utiliza las combinaciones externas (*OUTER JOIN* o *FULL*

OUTER JOIN). *LEFT JOIN* o *LEFT OUTER JOIN* y *RIGHT JOIN* o *RIGHT OUTER JOIN*, son subdivisiones de este procedimiento en que se mantiene todos los registros de la primera tabla mencionada o de la segunda respectivamente.

Si se aplica *LEFT JOIN* a las tablas de ejemplos el resultado será una relación en donde se encuentran todos los estudiantes incluso aquellos que no corresponden con un curso, los datos de la tabla derecha quedarían nulos si no poseen relación con la tabla de la izquierda. La sentencia sería:

*SELECT * FROM estudiante LEFT JOIN curso ON*

estudiante.id_curso = curso.id

Tabla 8: LEFT JOIN estudiante y curso

estudiante.nombre	estudiante.id_curso	curso.id	curso.nombre
Juan	51	51	Matemáticas
Madison	51	51	Matemáticas
Gaspar	54	54	Biología
Samantha	56	NULL	NULL

Con un *RIGHT JOIN* el resultado es otra relación en donde están presentes todos los registros de la segunda tabla, este caso curso, y los datos de la primera tabla que no poseen relación con la segunda son marcados con *NULL*.

*SELECT * FROM estudiante RIGHT JOIN curso ON*

estudiante.id_curso = curso.id

Tabla 9: RIGHT JOIN estudiante y curso

estudiante.nombre	estudiante.id_curso	curso.id	curso.nombre
Juan	51	51	Matemáticas
Madison	51	51	Matemáticas
NULL	NULL	52	Física
Gaspar	54	54	Biología
NULL	NULL	55	Literatura

Por último para obtener todos los registros de las dos tablas involucradas sin importar que todos los registros de una tengan relación con la otra se utiliza *FULL OUTER JOIN.*

*SELECT * FROM estudiante FULL OUTER JOIN curso ON*

estudiante.id_curso = curso.id

Tabla 10: FULL OUTER JOIN estudiante y curso

estudiante.nombre	estudiante.id_curso	curso.id	curso.nombre
Juan	51	51	Matemáticas
Madison	51	51	Matemáticas
NULL	NULL	52	Física
Gaspar	54	54	Biología
Samantha	56	NULL	NULL
NULL	NULL	55	Literatura

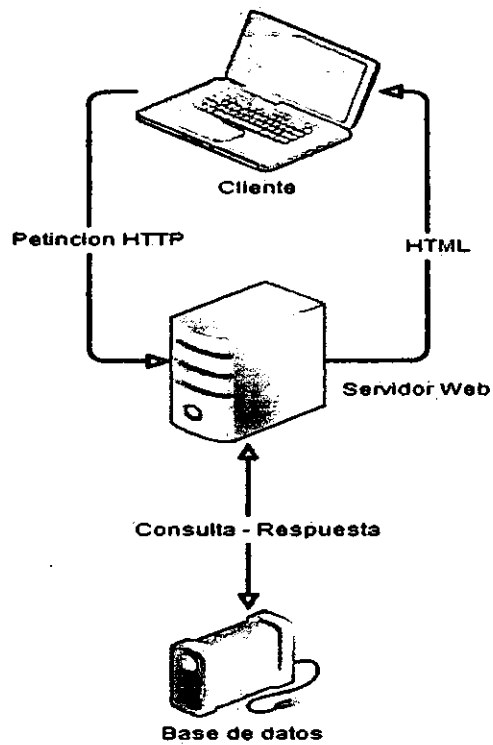
Todas estas operaciones de reunión son parte de SQL-92[27], como se observa cada variante de reunión está formada por las tuplas involucradas y una condición entre los atributos de las tablas. Además de la igualdad se puede usar el menor que (<), mayor que (>) para asociar las relaciones.

Como se mencionó antes los motores de base de datos soportan las funciones del estándar SQL pero también agregaron funciones propias. Si en una aplicación de software se necesita cambiar el motor de base datos, conlleva a rehacer la mayoría de las consultas para que se adapten a dicho motor. Como estrategia se desarrollan librerías que abstraen las consultas SQL en objetos del paradigma de programación orientada a objetos que son capaces de generar las ordenes adecuadas a al motor que se configure.

5.11 Aplicaciones WEB

Los sistemas basados en web hacen posible que se pueda obtener el recurso de la aplicación en cualquier lugar con acceso a internet. Su funcionamiento se basa en el protocolo HTTP de TCP/IP. El cliente envía peticiones a un servidor y este le responde con un archivo de texto que el navegador lo adapta visualmente al usuario (Gráfica 7). El archivo de texto comúnmente contiene un lenguaje de marcas llamado HTML (*HyperText Markup Language, Lenguaje de Marcado en español*).

Gráfica 7 Arquitectura de funcionamiento de una aplicación Web



Una aplicación web se diferencia significativamente de las típicas aplicaciones de escritorio porque hace uso intensivo de la red, reside en ella y da soporte a las necesidades de varios clientes, Se actualiza constantemente, por lo que no presenta con regularidad versiones planificadas. También presentan una estética adaptable con mucha facilidad [28].Sin embargo para mantener confidencial los contenidos se deben implementar fuertes políticas de seguridad en la infraestructura lógica y física [27].

A nivel lógico se pueden identificar muchas medidas de seguridad que se implementaran para evitar ataques web como:

Inyección SQL: consiste en la instrucción de código dentro de una petición de consulta del programa que puede ser, un cuadro de envío de información, un *login*, variables *URL*, un cuadro de búsqueda y su finalidad es ejecutar acciones que se envían a través de esos códigos[29]. Por ejemplo si una consulta de un campo de login es:

SELECT id FROM usuario WHERE nombre = " juan" AND contrasena=" clave "

Se puede hacer inyección insertando en su campo de usuario el siguiente código:

" or 1=1 - -

Quedando

SELECT id FROM usuario WHERE nombre = "" or 1=1 - - '" AND contrasena="
clave "

De esta forma la consulta solo validara la clausula 1=1 y los simbolos - - hacen que se omita la validación de la contraseña. De esta manera se buscara el primer usuario en una consulta y este saldrá en la petición, de esta manera no solo se puede incluir código de *login* también se pueden incluir códigos de modificación de datos, borrarlos, copiarlos, etc.

Como evitar este inconveniente, en cada campo se validara que la inclusión de caracteres como, punto, coma, comillas, paréntesis, y demás. Sean incluidos dentro del texto plano que se va a consultar para no altera el orden de la sentencia, o simplemente al encontrar esos caracteres se invalide la consulta.

Cross-site scripting: este tipo de ataque consiste en la inclusión de un pequeño código script el cual puede crear muchas falencias a pesar de las medidas de seguridad tomada, en esta se pueden crear formularios nuevos para robar contraseñas [30], copiar las *cookies* de sesión de un usuario, generar un mal comportamiento del sistema, fallos etc. Por ejemplo si se coloca una función script en un formulario, cuadro de texto, comentario, etc. se puede inducir a los usuarios del sistema a que realicen una acción, la cual no es la adecuada:

<script> alert (' anuncio ')</script>

Si dentro de un cuadro de mensajes se puede ejecutar el script se puede poner un texto como el siguiente.

"para verificar su cuenta, debe confirmar su contraseña".

Y crear un script el cual se cree un formulario y lo envíe a otra página, correo, etc., con lo cual la seguridad impuesta del sistema fallaría debido a que no tiene control de la

actividad que ha generado la nueva inclusión de un script de alerta y luego un formulario, los cuales no son parte del software.

Para evitar este tipo de ataques se debe verificar que los caracteres punto, coma, mayor que, menor que, no sean tomados como parte del código sino como un texto dentro de la pagina de esta manera al escribir un código script solo aparecerá como parte del contenido.

Cross-Domain Actions: consiste en la ejecución de comandos dentro del servidor del programa, haciéndose pasar por un usuario del sistema, aquí si un usuario del sistema envía un formulario, dicho formulario puede ser alterado en el trascurso o se puede enviar un formulario idéntico haciendo parecer al sistema que es el usuario original, lo cual lleva a genera, desconfianza en los usuarios por acciones no realizadas por ellos, robo de cuentas, entre otras.

Este método web es utilizado cuando los atacantes hacen copia de las cookies dentro de los navegadores o dentro de los clientes activos, y servidores. Incluso incrustar códigos *java script* y enviar información a los contactos haciéndose pasar por el usuario en cuestión, ejemplo:

```
<script type="text/javascript">
$.post("http://banco.com/modulo-tranferencias-bancarias.php",
{ monto: "10000", to_numero_de_cuenta: "1234" })
</script>
```

Aquí si no se validad la cookie un usuario que obtenga la cookie reciente de un cliente del banco, puede enviar una petición por el script y este será ejecutado debido a que la cookie aun se encuentra activa dentro del servidor.

Este método se puede evitar verificando la confiabilidad del destino, con lo cual se le colocan unos *token* a la sesión cookie por usuario y es comprobada en el servidor donde se realizara la petición quien luego decide si es válida o no. Debido a esto es obvio que el éxito de una aplicación web radica en un equilibrio entre funcionalidad y seguridad. La falta de pruebas de infiltración en un desarrollo de software lleva a que en cualquier

momento el sistema colapse por explotación de alguna debilidad que no fue corregida por negligencia.

5.12 Plataforma Android

Android es una plataforma libre basada en java creada para dispositivos móviles como celulares o tabletas. Fue desarrollada por la compañía Android Incorporated, que fue comprada por Google en el año de 2005 (Android, 2011). Su estructura la compone principalmente de un Framework de Java que corre en la maquina virtual Dalvik, siendo este último el corazón del sistema operativo, los datos los almacena en la base relacional SQlite . En el 2010 Android lideró el mercado de los móviles en Estados Unidos y sigue creciendo fuera de ese territorio («Android hits top spot in U.S. smartphone market | Wireless - CNET News», 2010).

6 METODOLOGÍA

La metodología utilizada es mixta puesto que se realizó trabajo de campo con la empresa para obtener información del funcionamiento del negocio, y por otro lado, se llevo a cabo una investigación bibliográfica con el fin de obtener conocimientos técnicos que contribuyeran a la presente investigación.

De ahí que, en el presente trabajo de grado se realizara un estudio del entorno actual del área de mercadeo y venta de forma general dentro de la empresa, donde se definieran los procesos directamente relacionados que afectan la toma de decisiones. Luego, se llevó a cabo un análisis de los datos que intervienen en el proceso, determinando cuáles de ellos son internos y externos para la organización, de ahí se extrajeron las características que sirvieron para el análisis por medio del algoritmo desarrollado. Seguidamente se planteó un modelo que muestra los resultados sometidos a analisis para su aprobación y corrección. Para llevar a cabo el proyecto se tuvo en cuenta los pasos de la metodología CRISP-DM propuesta en («CRISP-DM - Process Model», s.d.):

Comprensión actual del negocio:

Para la creación del modelo a seguir dentro de la investigación se analizaron las características de las empresas del mercado de forma general para estimar que variables del entorno del área mercadeo y ventas que serian utilizadas para el análisis del software. Para respaldar el proceso se obtuvieron datos a partir de visitas y entrevistas a los vendedores de empresas de productos de frutas, verduras, juguetes y ropa. GRAFICO,

Posteriormente, se determinaron las características presentes en cada una de las respuestas que se usaron para determinar que variables del entorno afectan directamente la decisión de compra de un articulo.

Durante la recolección de la información y como profundización del tema se identificaron los modelos de toma de decisiones(Cabeza de Vergara & Muñoz Santiago, 2004), que se estaban presentando en las empresas, y se observo que muchas de estas utilizaban modelos como el modelo racional económico, donde se estudian todas las

posibilidades de compra que más le favorezca a la empresa y se escoge la mejor. Otros de los modelos importantes fue modelo racional limitado, ya que muchos de los vendedores escogen una solución "x", que sea adecuada para el problema pero sin hacer un análisis de los otros métodos dispuestos del mercado.

Finalmente, para culminar esta etapa se creó una lista de las posibles características y las variables que afectan al proceso de decisión.

Comprensión de los datos:

Dentro de la elaboración del modelo de datos para el software se utilizaron los datos presentes en la empresa de juguetería *PARTY TIMES & DEKO LTDA*, en la cual se identificó inicialmente el modelo de toma de decisiones usado, en donde para un producto se selecciona una categoría y características coherentes luego, se analiza cuales de esos productos con características similares han sido vendidos previamente y se busca una estrategia similar de ventas; así para la persona encargada de la venta se asignan metas expuestas en cantidad y tiempo. Este proceso es apoyado por publicidad y promociones por medio de otros productos.

Para esta etapa de la investigación como en la mayoría de las empresas, el proceso de elección de un producto, se determina por el producto que más se vende, sin embargo, es importante analizar el impacto del producto que es más vendido y si este es vendido de forma individual o colectiva. Así una relación directa entre los artículos del almacén puede mejorar la expectativa de ingresos para inversión de capital en una característica del producto.

Con esto se decidió analizar para una venta los productos asociados a la venta y las características que intervienen en cada artículo de la venta.

Preparación de los datos:

Se realizó una selección de cada uno de los atributos de los datos obtenidos en el proceso anterior para su posterior análisis dentro del algoritmo desarrollado y el algoritmo Apriori, esperando que estas características sean relacionales y de importancia para el estudio obtenido en la actividad Comprensión Actual del Modelo.

Para elegir la mejor técnica de minería de datos dentro del tipo de estudio se partió de los datos del paso anterior y se analizó con los objetivos propuestos así, luego se hizo comparación con los trabajos (Cabeza de Vergara & Muñoz Santiago, 2004) y (Danger & Berlanga, 2001), con esto se escogió el algoritmo que favorecía a la creación de reglas de decisión como método de fortalecimiento, para los administradores y vendedores, el cual le permitiera de forma clara identificar los productos características del proceso de toma de decisiones.

Modelado:

Se dispuso los datos de manera organizada para los algoritmos, para mejorar su rendimiento y velocidad, se analiza su comportamiento dentro del proceso, por medio de comparaciones entre el algoritmo creado durante el modelo y el algoritmo original A priori. Para este proceso se analizarán: el costo computacional y la velocidad de respuesta.

Evaluación:

Para cada etapa del proceso se dispone de un seguimiento del programa, el cual será evaluado por medio de pruebas de individuales y pruebas en el entorno del desarrollo del trabajo; así se mantendrán pruebas unitarias a cada fase desarrollada verificando que se cumpla el objetivo propuesto para cada una de ellas.

Se estimaron las posibles correcciones del caso y se presentaron soluciones pertinentes con los objetivos de este proyecto.

Resultados del Modelo:

Se realizó un despliegue del modelo en su entorno, verificando principalmente las funciones en un entorno real. Se disponen realizar los últimos ajustes para la presentación de las reglas de apoyo a las decisiones y la comprensión con posibles usuarios.

7 RESULTADOS

7.1 Identificación de características para el modelo de toma de decisiones

De acuerdo con el desarrollo de la investigación en las empresas locales se realizó un análisis al personal de empresas del sector de Cartagena identificando los siguientes patrones al momento de tomar una decisión:

- Se usa un modelo racional económico y la decisión que se obtiene es analizada por un supervisor de ventas que tiene en cuenta diferentes alternativas.
- Se observo que el precio y la novedad del artículo es importante al momento de realizar una nueva compra.
- Para realizar el pedido y compra de los artículos se tiene en cuenta el nivel de inventario con un seguimiento del artículo más vendido.
- El software es usado para identificar de manera rápida los productos ausentes y más vendidos de acuerdo a la cantidad.
- El análisis de tendencia de productos se realiza mirando el historial de ventas de los meses y observando los productos más vendidos.

De esta forma un proceso de toma de decisiones en las empresas de ventas se lleva a cabo durante un proceso racional, se obtienen alternativas y problemas de venta, se realiza un estudio por parte de asesores y expertos de ventas para determinar la alternativa económicamente más viable y que favorece a su mercado, durante este proceso se tiene en cuenta los productos más vendidos e historial de la tendencia durante los periodos anteriores.

Con la anterior se pudo sintetizar características que afectan los procesos de decisión como son:

- Novedad del producto: para determinar esto se hizo un análisis en el mercado el producto de moda preferido por los compradores, dentro de esto se escogen entre productos para niños y niñas.
- Edad: se determina qué tipo de mercado puede usar el producto, pero también sirve para determinar preferencia de ellos.

- Marca: identifica una marca reconocida del producto que se está vendiendo.
- Precio de venta: influye en los compradores para decidir que producto usar, también puede afectar las promociones que esta conlleva.
- Productos Complementarios: se muestra en que un producto puede ser usado por otro artículo
- Presentación: la presentación del artículo los colores, tamaño, flexibilidad, diferentes usos.

De acuerdo con lo anterior se realizo un análisis de datos de ventas de la empresa *PARTY TIMES & DEKO LTDA*, estudiando las relaciones y atributos de los productos concluyendo con las reglas de decisión generadas, para hacer esto se tuvo en cuenta las características previamente mencionadas. Este proceso de análisis se explicaría con mayor detalle más adelante.

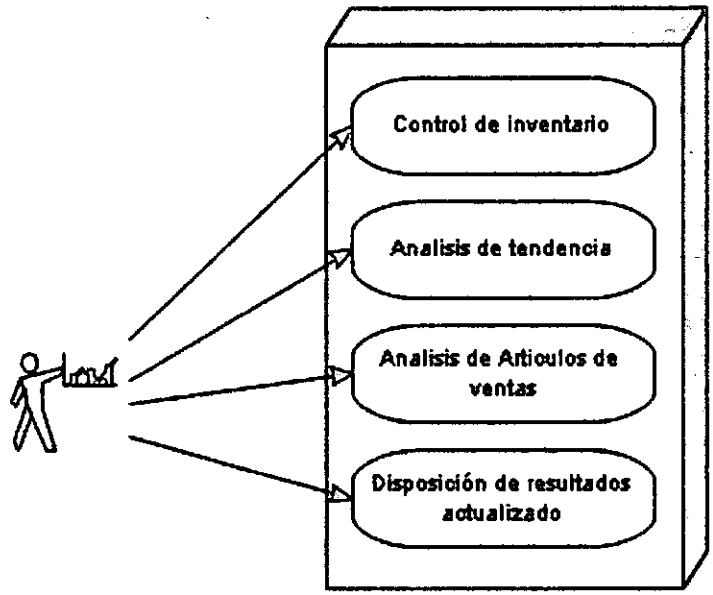
7.2 Marco de arquitectura del software

7.2.1 Requerimientos del sistema

Luego de la obtención de los modelos de toma de decisiones y las variables que interviene en el proceso se procedió a determinar cuáles eran los requerimientos que debería tener un sistema de apoyo a las decisiones dentro de una empresa del entorno.

Se creó un diagrama de requerimientos para el sistema de software (ver Gráfica 8).

Gráfica 8 Diagrama de casos de uso y Requerimientos del sistema



Control de inventario: dentro de este proceso el sistema deberá tener un control del inventario para identificar cuales productos se necesitan dentro de la empresa, cuales son los productos que se tienen actual mente y la cantidad de productos que se han vendido.

Análisis de tendencia: el sistema deberá determinar en cierto periodo de tiempo cual es la tendencia de los productos a comprar por los clientes, que artículos se comprarían, y que artículos compran en conjunto con más frecuencia los clientes.

Análisis de artículos de ventas: el sistema debe ser capaz de analizar los artículos que se relacionan en las ventas, cuales son los que más se venden, que características tiene el articulo más vendido y con esto ayudar a los encargados a determinar cómo se puede mejorar la venta de ese producto o de los productos relacionados.

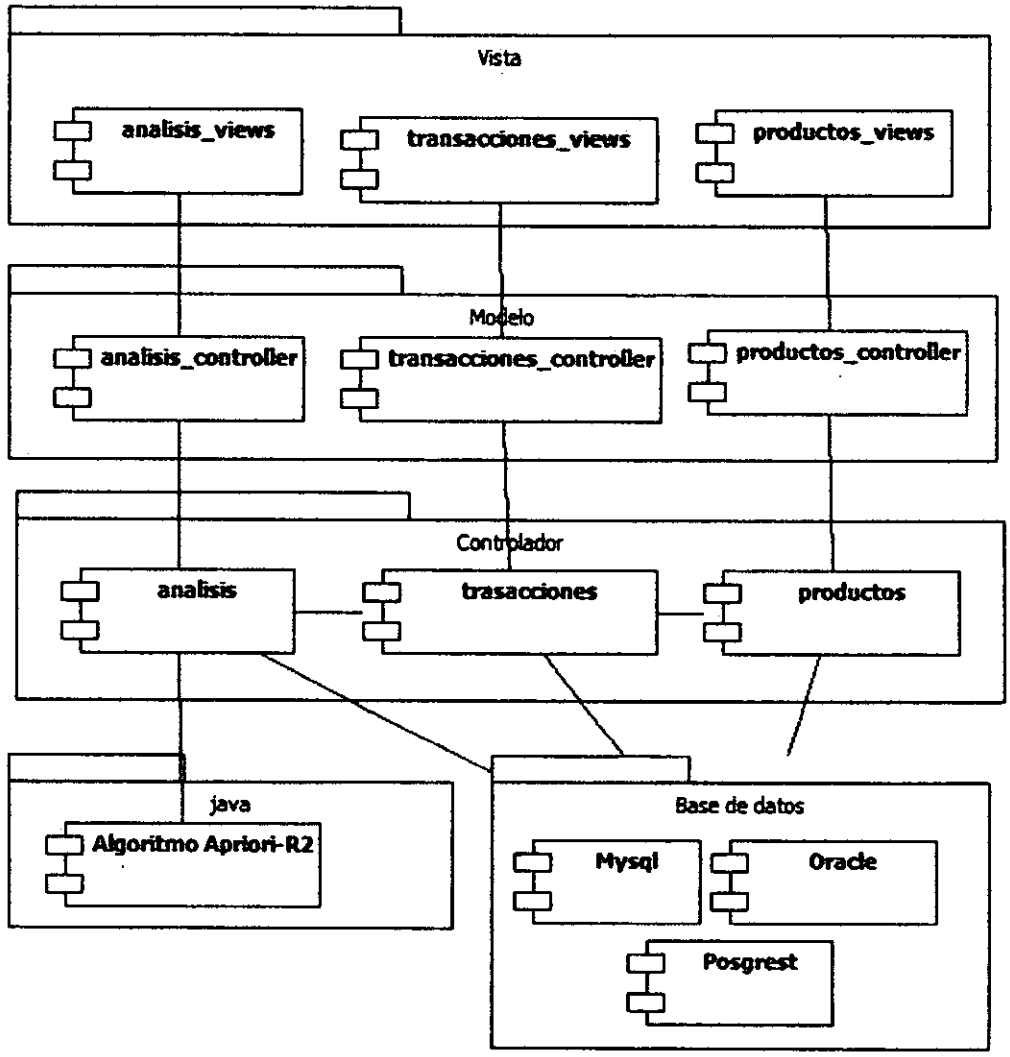
Disposición de resultados actualizados: el sistema debe estar actualizado para determinar tendencias que sucedan en el momento, con la posibilidad de dar una herramienta necesaria para el encargado al momento de tomar una decisión de compra o

de venta. También se hace necesario disponer de esta información en cualquier lugar o realizar análisis en periodos aleatorios, de acuerdo a la necesidad del usuario.

7.2.2 Vista de desarrollo del software

Dentro de este diagrama se encontrara los módulos, partes del software y la estructura de cómo se encuentran empaquetados, estos son: base de datos, programas, componentes (Gráfica 1Gráfica 9).

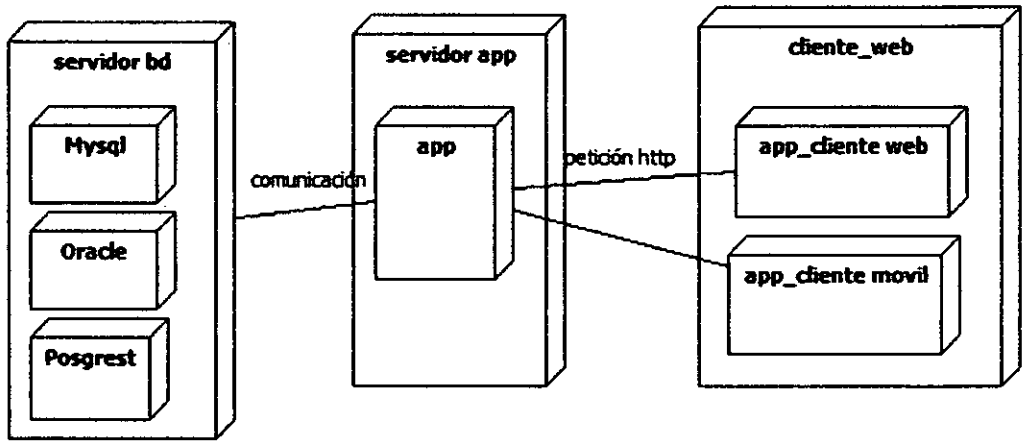
Gráfica 9 Diagrama de componentes



7.2.3 Vista de despliegue

Este diagrama representa la vista física del software donde se ve su escalabilidad (ver Gráfica 10).

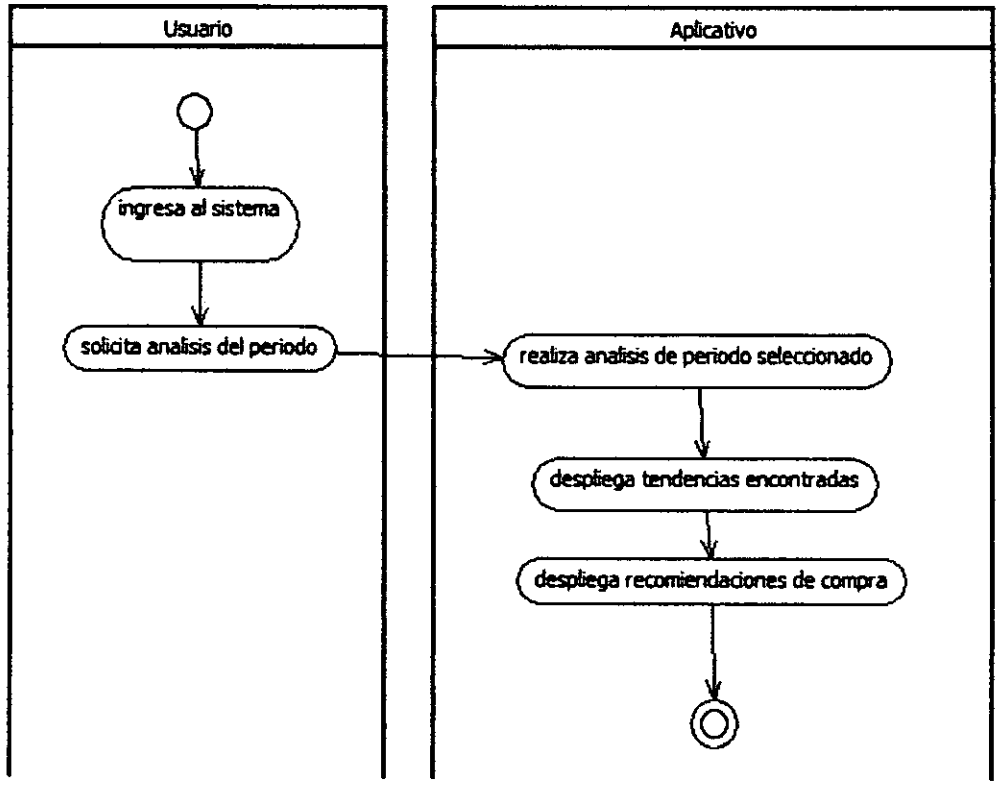
Gráfica 10 Diagrama de despliegue



7.2.4 Diagrama de actividades

Dentro de este diagrama se puede ver las actividades que se realizan en el sistema de acuerdo a los requerimientos funcionales de la clase análisis (Gráfica 11).

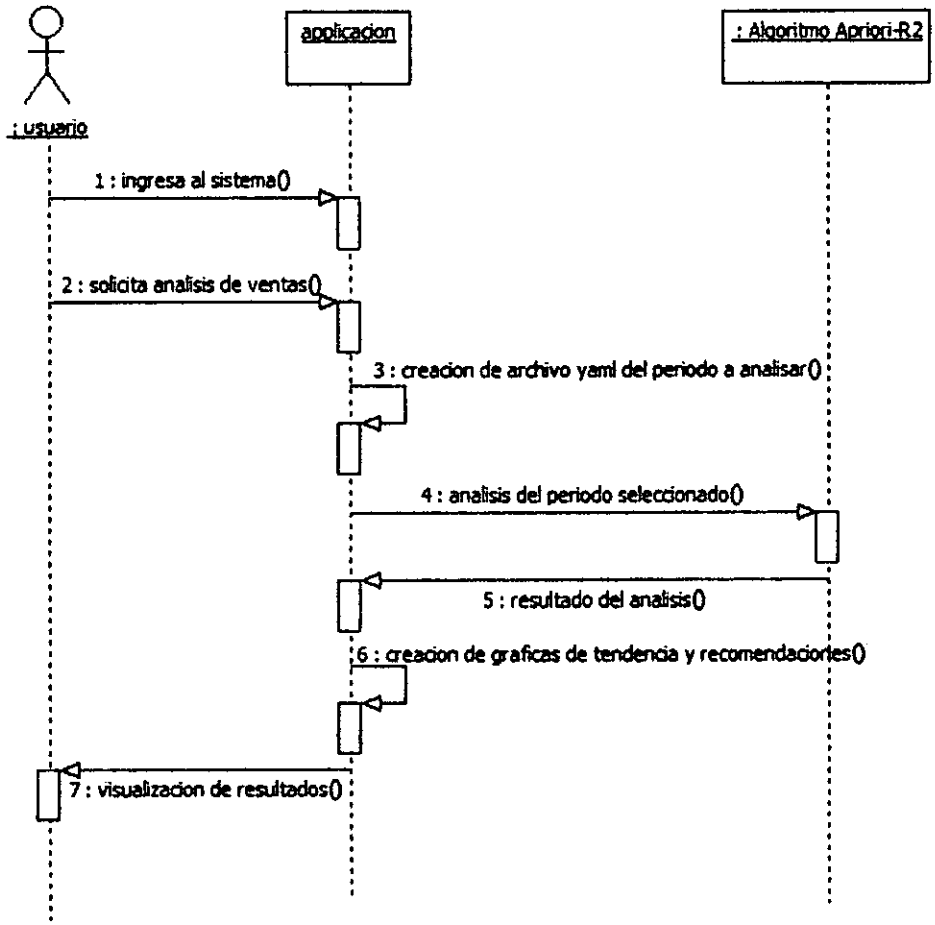
Gráfica 11 Diagrama de actividades



7.2.5 Diagrama de secuencia

Este diagrama describe la secuencia para obtener el análisis de las ventas por parte del sistema (ver Gráfica 12).

Gráfica 12 Diagrama de secuencia



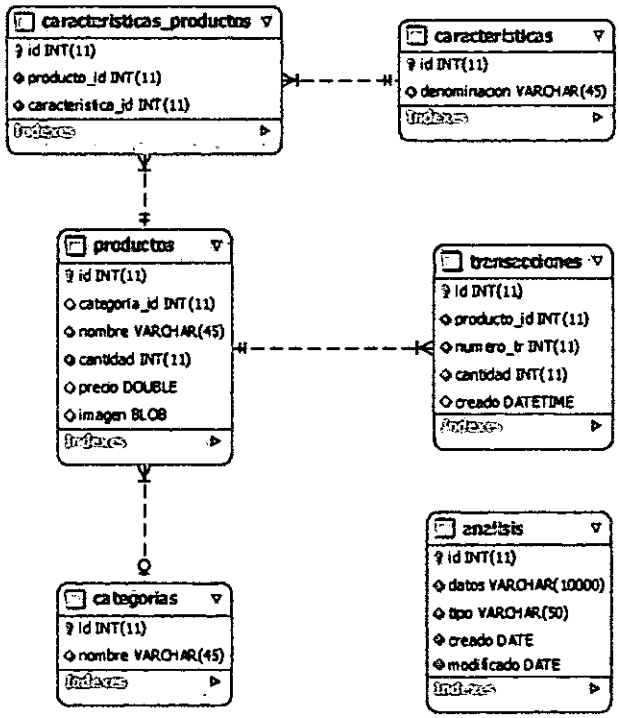
7.2.6 Creación de estructura lógica de las características a analizar

La investigación utilizó datos de ventas de la empresa *PARTY TIMES & DEKO LTDA,* para poner a prueba el software con datos reales y dar veracidad su eficiencia .

Al momento de extracción de los datos de la empresa, se encontró que estos eran ingresados en archivos Excel, otros en el programa de contabilidad Trident y los registros de ventas actuales estaban en sus facturas.

Con esto se procedió al desarrollo y diseño un modelo de base de datos para que los datos fueran utilizados por el software, la Gráfica 13 muestra este esquema:

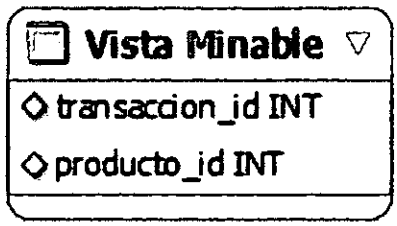
Gráfica 13 Estructura lógica de la Base de datos



Este modelo propuesto se organiza en tablas que representan los productos, su relación con las características y los registros de ventas de cada transacción, necesarios para los objetivos del proyecto.

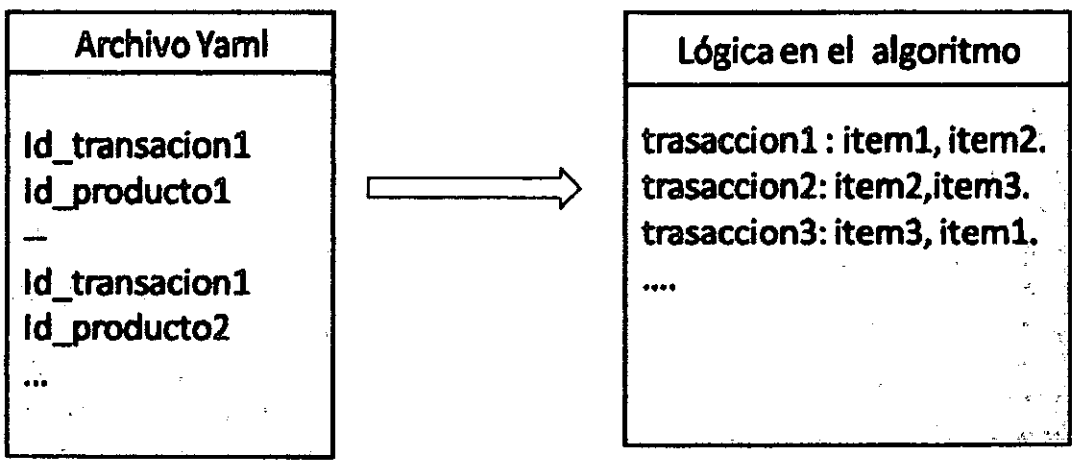
Luego de haber diseñado esta estructura, se identificaron los datos para el posterior análisis como las transacciones y los productos en las ventas. Los datos necesarios se agruparon en una sola tabla (vista minable) para ser minados (Gráfica 14):

Gráfica 14 Representación de la vista minable



Tomando de base a esta última representación de los datos se diseñó una estructura en formato YAML, en la que se agrupaba los identificadores de cada transacción y un producto esto dentro de la lógica del algoritmo representaría los productos que poseería cada transacción (ver Gráfica 15):

Gráfica 15 Estructura archivo YAML



7.3 Estudio de Requerimientos para elección de algoritmo

Para la elección del método de minería de datos se verificaron las siguientes indicaciones las cuales son necesarias para cumplir los objetivos de la investigación:

- Se necesitan encontrar conjuntos frecuentes entre los datos
- Identificar relaciones entre productos
- Poder identificar un nivel de confianza entre las relaciones

- Generación de reglas de decisión.
- Análisis la mayor y menor cantidad de relaciones entre productos de los datos analizados.
- Al momento del análisis de los conjuntos analizados no se excluyan resultados o se descarten ítem.
- Poder analizar de manera eficaz una gran cantidad de datos.

Por ello y con el estudio del trabajo "Aproximación al proceso de toma de decisiones en la empresa barranquillera" (Cabeza de Vergara & Muñoz Santiago, 2004) y "Búsqueda de Reglas de Asociación en bases de datos y colecciones de texto" (Danger & Berlanga, 2001), se escogió utilizar el algoritmo de Apriori.

7.4 Estudio de cambios y mejoras para el algoritmo.

El algoritmo de Apriori se dispuso de la siguiente forma para mejorar su rendimiento:

Recorre el archivo dispuesto por la base de datos para analizar los datos correspondientes, los guarda en memoria, luego realiza el primer recorrido identificando los elementos frecuentes y seleccionando los conjuntos que pertenecen al soporte dado.

Dentro de este paso los ítem que no se encuentren en los dados por los frecuentes será descartado del análisis, así se reducen las transacciones. Luego el algoritmo procede a seguir con los análisis sobre el ítem frecuentes, buscando identificar los elementos candidatos dentro de los subconjuntos.

Para garantizar que se mejorara la eficiencia del algoritmo se modificó la estructura de recorrido para que este no recorriera dos veces un subconjunto de ítem en la que se está seguro que no ayudará a obtener ítems frecuentes, facilitando la identificación de los datos frecuentes con menos recorridos sobre la base de datos de transacciones.

La lógica que minimiza el recorrido se realizó de la siguiente forma:

El algoritmo marcaba las transacciones que se tenía la seguridad que ya no serían útiles para obtener el soporte del siguiente conjunto de ítems. Las reglas que rigen que

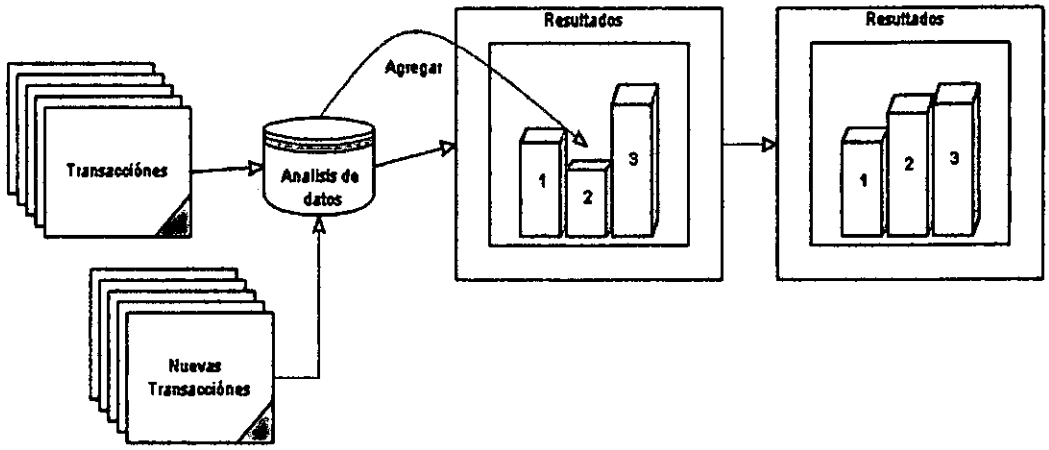
transacciones debían ser excluidas en el momento que se calculaba el soporte de un conjunto de ítems candidatos eran los siguientes:

- Los registros que poseían un solo elemento a la vez que se estaba calculando el soporte de 1-itemset.
- Los registros que tenían los mismos elementos que los conjuntos de ítems candidatos actuales y ambos tenían la misma cantidad de elementos.
- Los registros cuya cantidad de elementos era menor que la cantidad que poseía el conjunto ítems candidatos estuviera siendo analizado en ese momento.

Para mejorar el análisis de las transacciones dentro del algoritmo APriori se ha optado por serializarlas en un archivo YAML, cada registro se ha representado por el identificador de la dicha transacción y el identificador del elemento siguiendo el modelo de la estructura de "formato transaccional" puesto que es más cómodo para minar (Hornick et al., 2007, pág 94). El algoritmo Apriori desarrollado realiza el minado en el archivo YAML en vez de hacerlo directamente en la base de datos, puesto que se ha demostrado mejor rendimiento guardando los datos en un archivo de texto (Sarawagi et al., 2000).

Por último se creó una estructura incremental la cual funciona de la siguiente manera: el algoritmo realiza un análisis de los registros en un determinado periodo y guarda los resultados, en caso de que se agreguen más transacciones dentro de ese periodo, estas sería detectadas por el algoritmo en un posterior análisis y no tendría la necesidad de analizar todos los datos, sino solamente los registros que han sido agregados recientemente, de esta manera los resultados serían agregados a los obtenidos en el primer análisis, como se puede apreciar en la Gráfica 16 :

Gráfica 16 Proceso de obtención de resultados



Para la implementación del algoritmo se escogió el lenguaje de programación Java por su practicidad de trabajar con objetos y su característica multiplataforma.

Teniendo en cuenta el lenguaje de programación, se desarrolló una variación del algoritmo Apriori el cual será nombrado dentro de este documento con el nombre de "Apriori-r2" (r de reducción de tiempo y 2 por ser una variación al Apriori original), con la características de que sus funciones para el minado de los datos pudieran ser accedidos por medio de la línea de comandos, para facilitar su posterior integración con la interfaz web del proyecto.

7.5 Comparación del algoritmo creado Apriori-r2 y el algoritmo original Apriori

Tan pronto se desarrolló el algoritmo el lenguaje de programación Java, se realizaron pruebas de rendimiento con los datos de ventas brindados por la comercializadora de productos.

El tiempo de respuesta se midió para diferentes valores de soporte tomado desde 20% hasta 100%. En todos estos rangos los valores fueron más altos en A priori (ver Tabla 11).

Tabla 11. Apriori vs Apriori-r2

	Tiempo mili-segundos	
Soporte (%)	Apriori	Apriori-r2
20	1301	968
30	601	212
40	571	170
50	527	135
60	535	137
70	548	137
80	584	147
90	523	141
100	523	136

En ambos casos la diferencia de tiempo es mayor a medida que el soporte es más alto, esto se debe a que se encuentran más ítems frecuentes, sin embargo Apriori-r2 se ejecuta más rápido que Apriori aunque no se encuentren ítems frecuentes, debido a que Apriori-r2 descartas las transacciones con un ítem en su primer recorrido posean o no ítems frecuentes (ver Tabla 12).

Tabla 12. Comportamiento ítems frecuentes

Soporte (%)	Ítems Frec.
20	13
30	8
40	0
50	0
60	0
70	0
80	0
90	0
100	0

Para ambos casos analizados y con las modificaciones hechas se evidenció que los 2 algoritmos obtuvieron la misma cantidad de ítem frecuente y no fue omitido ninguno durante el proceso.

También se analizaron dichos algoritmos en tres casos específicos:

- Muchos productos y pocas transacciones. (escenario a)
- Cantidad de transacciones y productos similares. (escenario b)
- Poca cantidad de productos y muchas transacciones. (escenario c)

Para obtener estos escenarios se tuvo que eliminar transacciones ó productos de la base de datos modelo. En los tres escenarios Apriori-r2 también presentó un mejor rendimiento que Apriori; en algunos casos fue aproximadamente 40 veces más rápido tal como se evidencia en el "escenario b" (ver Tabla 14), en el "escenario c" la relación se mantuvo 1 a 4 para cada soporte (ver Tabla 15); en el "escenario a" el mejor rendimiento fue aproximadamente de 20 veces (ver Tabla 13).

Tabla 13 Escenario a. (239 productos y 142 transacciones)

Apriori (tiempo ms)	Apriori-r2 (tiempo ms)	Soporte (%)	Ítems frecuentes
625	375	20	7
359	94	30	5
344	64	40	1
344	47	50	0
944	47	60	0

Tabla 14 Escenario b (239 productos y 265 transacciones.)

Apriori (tiempo ms)	Apriori-r2 (tiempo ms)	Soporte (%)	Ítems frecuentes
500	219	20	7
422	141	30	2
421	93	40	1
423	94	50	1
468	125	60	0

Tabla 15 Escenario c (58 productos y 568 transacciones)

Apriori (tiempo ms)	Apriori-r2 (tiempo ms)	Soporte (%)	Ítems frecuentes
547	219	20	5
500	156	30	2
500	141	40	2
469	125	50	1
469	125	60	1
484	125	70	0

También se hicieron pruebas variando el valor de la confianza. Para ello se tomaron los datos de ventas originales y se omitieron los escenarios anteriores debido a que en estos no se encontraban reglas de asociación. Se probaron los valores de confianza para diferentes soportes, aquellos que generaban reglas y los que no, y se concluyó que en los dos algoritmos el rendimiento es prácticamente el mismo (ver Tabla 16, Tabla 17, Tabla 18 y Tabla 19), menor cuando el soporte es mayor (ver Tabla 16, Tabla 17, Tabla 18 y Tabla 19) e independiente del número de reglas que encuentre (ver Tabla 16, Tabla 17). Los soportes tomados fueron los de 20% que generaba reglas y el de 30% que no para ningún valor de confianza.

Tabla 16 Variación de la confianza para Apriori con un soporte de 20%

Apriori				
Soporte (%)	Ítems Frec.	Confianza (%)	Tiempo (nano-segundos)	reglas
20	13	20	1.312.484	6
20	13	30	1.348.518	6
20	13	40	1.398.460	6
20	13	50	1.330.675	6
20	13	60	1.330.313	1
20	13	70	1.354.688	0
20	13	80	1.366.549	0
20	13	90	1.360.524	0

Tabla 17 Variación de la confianza para Apriori-r2 con un soporte de 20%

Apriori-r2				
soporte	Ítems Frec.	confianza	Tiempo (nano-segundos)	reglas
20	13	20	1.326.146	6

20	13	30	1.373.387	6
20	13	40	1.397.269	6
20	13	50	1.391.419	6
20	13	60	1.333.492	1
20	13	70	1.379.572	0
20	13	80	1.310.786	0
20	13	90	1.357.868	0

Tabla 18 Variación de la confianza para Apriori con un soporte de 30%

Apriori				
Soporte (%)	Ítems Frec.	Confianza (%)	Tiempo (nano-segundos)	reglas
30	8	20	370.979	0
30	8	30	326.234	0
30	8	40	355.952	0
30	8	50	377.322	0
30	8	60	363.632	0
30	8	70	314.416	0
30	8	80	558.304	0
30	8	90	302.381	0

Tabla 19 Variación de la confianza para Apriori-r2 con un soporte de 30%

Apriori-r2				
Soporte (%)	Ítems Frec.	Confianza (%)	Tiempo (nano-segundos)	reglas
30	8	20	344.933	0
30	8	30	357.288	0
30	8	40	340.592	0
30	8	50	370.978	0
30	8	60	352.946	0
30	8	70	377.657	0
30	8	80	356.954	0
30	8	90	364.634	0

En general el algoritmo desarrollado presentó un mejor rendimiento con respecto al algoritmo Apriori con respecto a la obtención de ítems frecuentes sin embargo en hallar las reglas de asociación es exactamente el mismo comportamiento, esto se debe a que se hallan después de tener los ítems frecuentes. El mayor rendimiento en la obtención de ítems frecuentes se obtuvo cuando las transacciones y los productos estaban

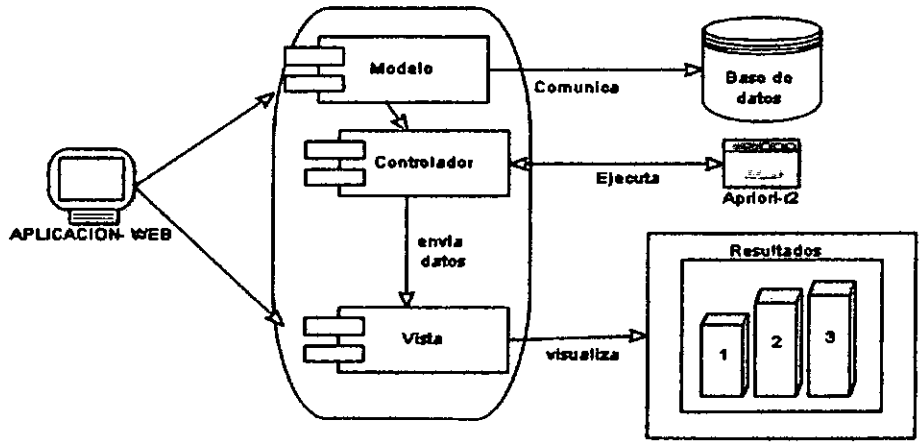
equilibrados. Para los casos extremos en los que se presentaban pocos productos o pocas transacciones los resultados revelaron que el comportamiento del rendimiento fue exponencial y constante para cada valor de soporte respectivamente.

7.6 Diseño del software en la web y adaptación con el algoritmo realizado

Una vez verificada el modelo de Apriori y su resultado se creó el diseño de la interfaz web para mostrar las reglas de decisión generadas.

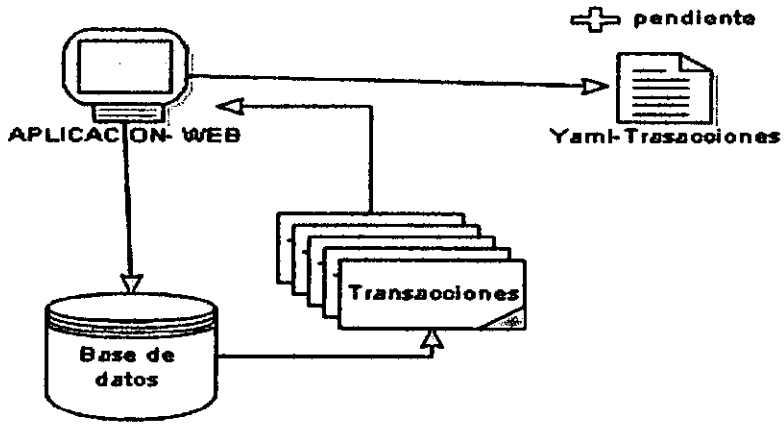
Para la creación del software web se utilizó el lenguaje de programación PHP, y se utilizó el esquema de Modelo-Vista-Controlador (MVC) para separar los procesos de acceso a la base de datos, la lógica del negocio y la presentación (ver Gráfica 17).

Gráfica 17 Esquema de la aplicación web



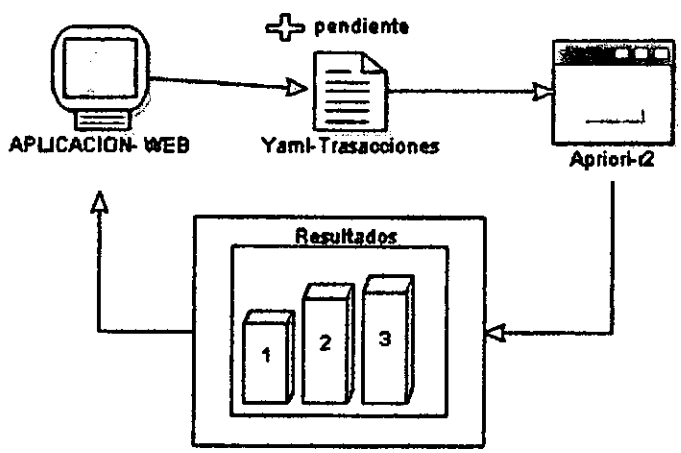
En el proceso se crearon las funciones para consulta de artículos y la creación de un archivo YAML, como se había mencionado. Para generar este archivo el programa hizo una consulta a los de ventas de la empresa identificando los productos que intervienen en ella y se dispuso apropiadamente en la estructura YAML para el proceso de minado (id de transacciones e id de producto en cuestión), la Gráfica 18 ilustra el proceso.

Gráfica 18 Esquema de generación archivo YAML en la aplicación web



Una vez generado los archivos YAML con éxito se creó el método desde un controlador web para llamar a la función del algoritmo, que serviría para leer la estructura de datos y realizar el proceso de minado, posteriormente generó un archivo de los resultados obtenidos (Gráfica 19).

Gráfica 19 Proceso de análisis de un archivo YAML

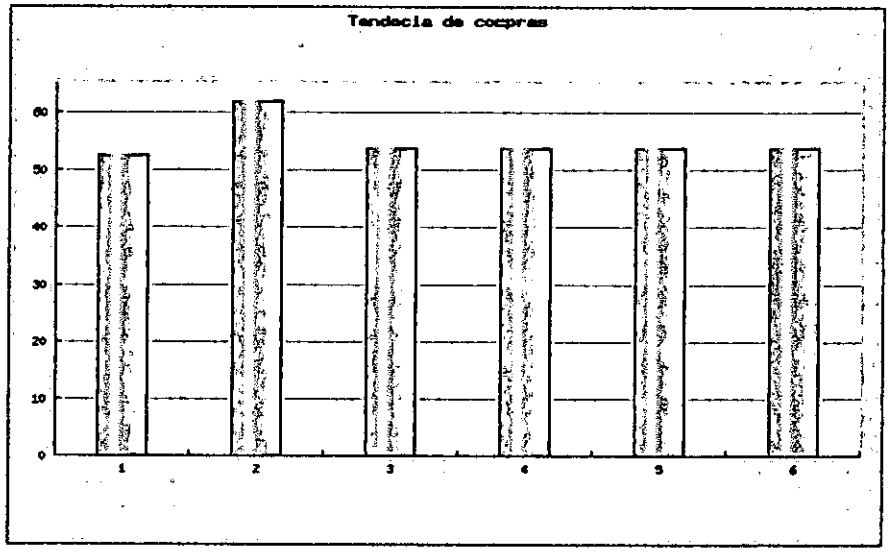


Una vez analizados los datos anteriores se creó una función dentro del programa para determinar cuáles fueron las tendencias encontradas, productos más frecuentes y reglas generadas.

Estos procesos fueron graficados y dispuestos en tablas, de forma automática por el sistema web (ver Gráfica 20):

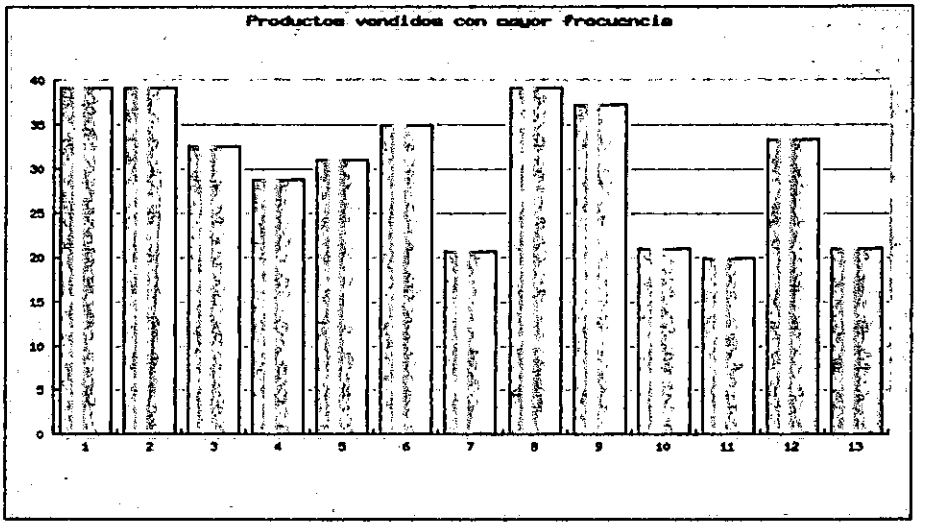
Gráfica 20 Tendencias encontradas

Tendencia de compras



Id	tendencias de compras
1	Por la compra de **"moto ben 10 friccion"** se llevaron **"helicoptero plane"**
2	Por la compra de **"helicoptero plane"** se llevaron **"moto ben 10 friccion"**
3	Por la compra de **"helicoptero mediano"** se llevaron **"moto ben 10 friccion"**
4	Por la compra de **"moto ben 10 friccion"** se llevaron **"helicoptero mediano"**
5	Por la compra de **"Reloj Ben 10"** se llevaron **"moto ben 10 friccion"**
6	Por la compra de **"moto ben 10 friccion"** se llevaron **"Reloj Ben 10"**

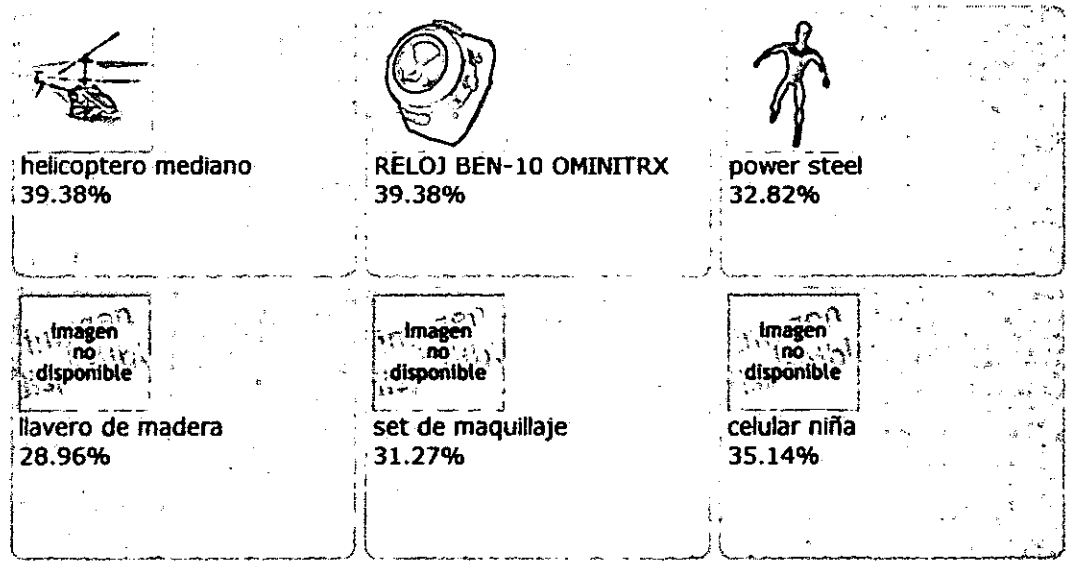
Articulos más Vendidos



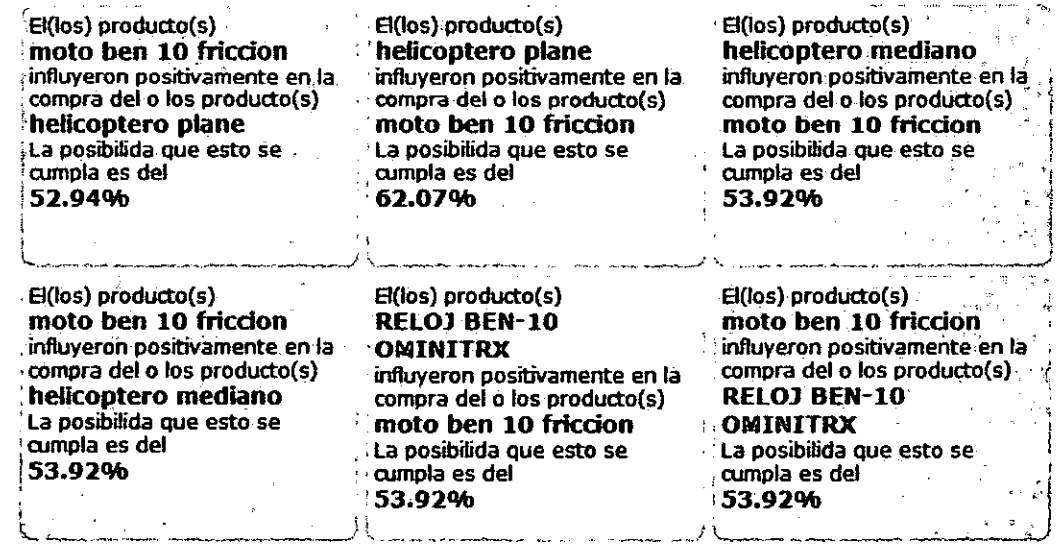
Id	Productos
1	helicoptero mediano
2	Reloj Ben 10
3	power stel
4	llavero de madera
5	set de maquillaje
6	celular niña
7	moto ben 10 friccion, helicoptero plane
8	moto ben 10 friccion
9	max steal power
10	moto ben 10 friccion, helicoptero mediano
11	carro montable
12	helicoptero plane
13	moto ben 10 friccion, Reloj Ben 10

Además de los anteriores resultados, la aplicación es capaz de mostrarlos de forma simplificada para resaltar solo la información de interés, que serían los productos que deberían comprar (ver Gráfica 21) y las tendencias utilizadas en el último análisis (ver Gráfica 22)

Gráfica 21 Recomendaciones de productos en la aplicación web



Gráfica 22 Tendencias de compras mostradas por la aplicación web

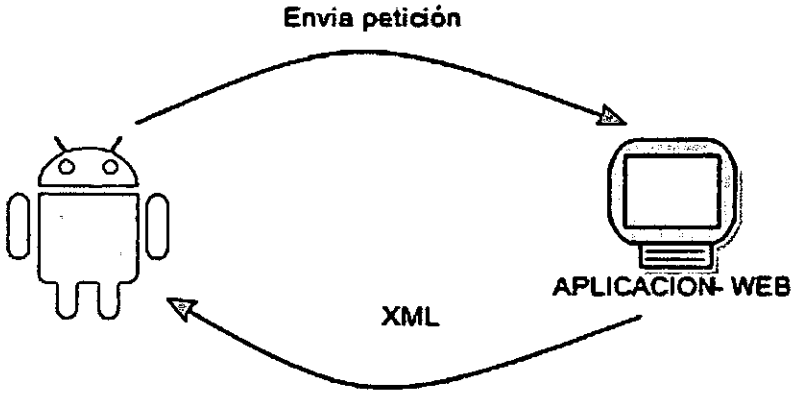


Con la creación de la plataforma web se logró el objetivo de adaptar el modelo del algoritmo realizado y los análisis a un sistema donde se muestra de forma clara los resultados y que estos pudieran ser representados por graficas para los posibles usuarios. En este punto del desarrollo del software se hizo notorio que el análisis de las tendencias de compras entre productos y sus características, que representarían el patrón de consumo, debían realizarse en un periodo de tiempo. Este motivo condujo a que el software fuera capaz de ejecutar periódicamente el análisis de las tendencias (generar el archivo YAML y ejecutar el programa java), esto se logró al combinar las tareas programadas del sistema operativo y habilitar las funciones de la aplicación por línea de comando.

7.7 Despliegue de software en dispositivos móviles

Con el desarrollo de la aplicación web se logró integrar y extender las funciones a los dispositivos móviles como celulares bajo la plataforma Android. Esta consta de una aplicación que realiza consultas al sistema de ventas y devuelve un XML con la información para se despliegue en la interfaz de usuario del móvil (ver Gráfica 23).

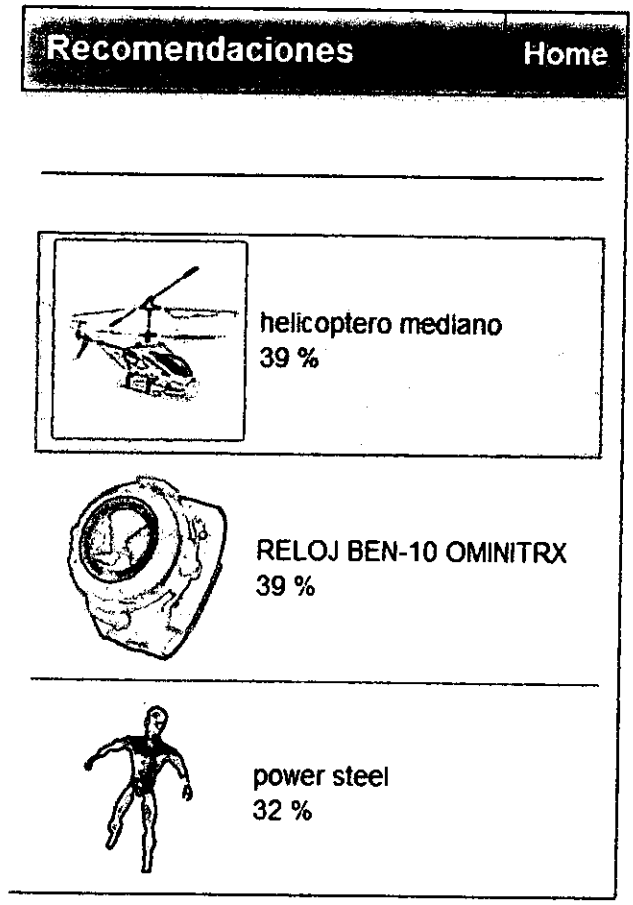
Gráfica 23 Comunicación Android con la Aplicación Web[2]



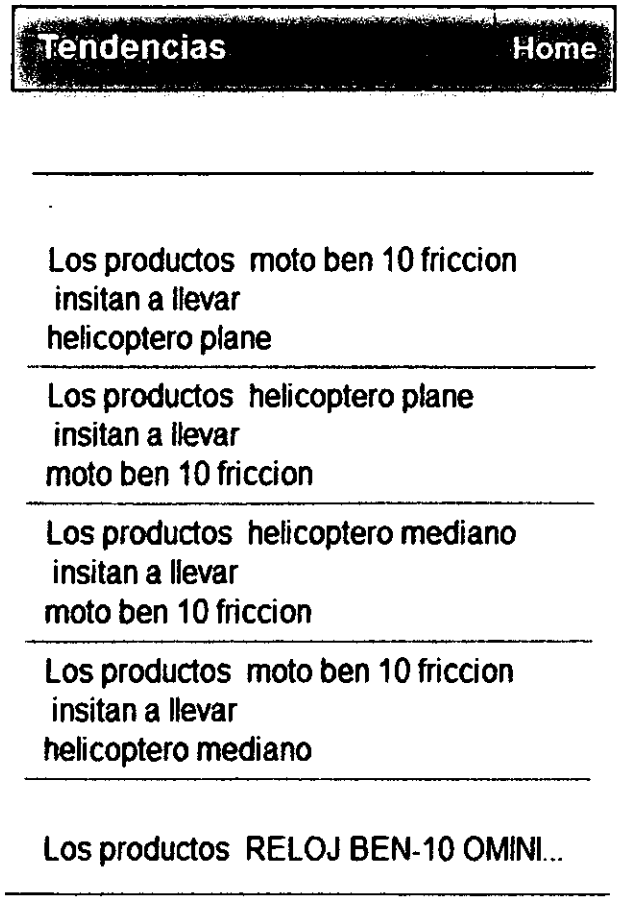
[2] Icono de Android es marca registrada de Google

La aplicación está desarrollada para consultar las existencias de los productos, informar cuales se deben tener en cuenta para realizar futuras compras (ver Gráfica 24) y muestra las tendencias de los clientes de forma similar a como lo hace la aplicación web (ver Gráfica 25).

Gráfica 24 Recomendaciones de productos en la aplicación móvil



Gráfica 25 Tendencias de compras mostradas por la aplicación móvil



7.8 Estudio del impacto del software dentro de una empresa.

A partir de los resultados obtenidos se presentó parte de los resultados a la empresa *PARTY TIMES & DEKO LTDA,* para determinar qué tan útiles fueron los análisis de los datos previamente analizados, de ahí se anotaron algunas observaciones de los usuarios al momento de interactuar con el software, mejorando los procesos de agregar las características presentes en los productos, la presentación de la reglas, para un mejor análisis.

En contraste con las reglas generadas y los inventarios presupuestados por la empresa se escogió una muestra donde se muestra que el 30 % de los artículos mencionados por el algoritmo debieron ser pedidos ya que su tendencia de compra estuvo dentro del rango analizado.

Esto demostró que en cierto punto el proceso de tendencia puede ser de gran ayuda para saber qué productos comprar y tener una alta probabilidad de cual producto será vendido dentro de un periodo específico. También se mostró que el programa respondió de manera satisfactoria con datos reales, lo cual significa que puede ser usado en cualquier tipo de negocio dedicado a la venta de productos.

8 CONCLUSIONES Y RECOMENDACIONES

Con los estudios realizados sobre los procesos de tomas de decisiones y los modelos encontrados se concluyó cuales son las variables que intervienen en el proceso de decisión para aplicarlas en el algoritmo.

Con las características y requerimientos del proyecto se logró desarrollar un algoritmo derivado del Apriori el cual mejoró los recorridos y sus tiempos de respuesta, también se desarrolló una estructura dinámica para la inclusión de nuevos datos y la migración de los datos a un archivo externo (YAML), mejorando el acceso a los datos y el tiempo de análisis. También se evidenció que el tiempo requerido para generar las reglas es demasiado menor que el necesario para obtener ítems frecuentes y que dicho tiempo solo varía dependiendo del número de ítems frecuentes que se encuentren.

La integración del sistema en un entorno web con el algoritmo desarrollado, resultó muy útil gracias a los métodos de acceso que se crearon en el algoritmo (archivo YAML, línea de comandos). De esta manera el sistema web logró ejecutar el algoritmo desarrollado en java en segundo plano por medio de comandos del sistema, concluyendo de este proceso que al momento de realizar un análisis la plataforma web no está sobrecargada debido a que los cálculos de ítems frecuentes y las reglas de asociación se realizan independientemente por la plataforma java del algoritmo.

La visualización de resultados como los gráficos, reglas de decisión, tendencias y el estudio de características tuvieron un impacto positivo en la empresa *PARTY TIMES & DEKO LTDA,* puesto que se pudo comprobar que este nuevo sistema de apoyo a las decisiones mostraba características que eran muy útil mostrándole como se podía aprovechar los productos para llamar la atención de los clientes, la posibilidad de estudiar con más detalle el inventario, los productos y la generación de tendencias en tiempo de venta le ayudó a la empresa a generar estrategias de ventas y fortalecer algunas que ya se tenía.

Como observación al momento de implementación del sistema puede existir en la empresa un software que disponga una estructura de base de datos diferente a la propuesta, debido a que en la mayoría de los casos estos son de marca registrada y en

ciertas ocasiones no es posible acceder a estos sistemas, lo cual dificultaría la integración de los modelos. Para esto se creará un archivo donde se encuentre los datos de las ventas y luego sea ingresado al sistema para realizar el análisis de todos sus datos. Como trabajos futuros se podría mejorar este proceso con un posterior estudio de los sistemas actuales y ver la disposición de los datos de ventas y registro de las mismas para lograr una adaptación entre los modelos y el modelo propuesto.

BIBLIOGRAFÍA

Acuna, E. (2010). Minería de datos: Regla de asociación. Recuperado a partir de http://math.uprm.edu/~edgar/asorulesacu.pdf_

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules, 487--499.

Agrawal, R., Tomasz, I., & Swami, A. (1993). Mining Association Rules between Sets of Items in Large Databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26-28, 1993* (Vol. 22, págs 207-216). New York, NY, USA.

Android. (2011). About the Android Open Source Project | Android Open Source. Recuperado Abril 27, 2011, a partir de http://source.android.com/about/index.html

Android hits top spot in U.S. smartphone market | Wireless - CNET News. (2010). . Recuperado Abril 27, 2011, a partir de http://news.cnet.com/8301-1035_3-20012627-94.html

Asghar, S., Fong, S., & Hussain, T. (2009). Business Intelligence Modeling: A Case Study of Disaster ManagementOrganization in Pakistan. *Fourth International Conference on Computer Sciences and Convergence Information Technology.*

Ben-KiKi, O., Evans, C., & döt Net, I. (s.d.). YAML Ain't Markup Language (YAML™) Version 1.2. Recuperado Enero 20, 2011, a partir de http://www.yaml.org/spec/1.2/spec.html

Bodon, F. (2003). A fast APRIORI implementation. *IN PROCEEDINGS OF THE IEEE ICDM WORKSHOP ON FREQUENT ITEMSET MINING*

IMPLEMENTATIONS. Recuperado a partir de http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.1158

Borgelt, C. (2003a). Efficient Implementations of Apriori and Eclat. *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementation.* Melbourne, Florida, USA.

Borgelt, C. (2003b). Recursion Pruning for the Apriori Algorithm. Department of Knowledge Processing and Language Engineering School of Computer Science.

Borgelt, C. (2005). An Implementation of the FP-growth Algorithm. *Proceedings of the 1st international workshop on open source data mining frequent pattern mining implementations - OSDM '05* (págs 1-5). Presented at the the 1st international workshop, Chicago, Illinois. doi:10.1145/1133905.1133907

Cabeza de Vergara, L., & Muñoz Santiago, A. E. (2004, Diciembre). Aproximación al proceso de toma de decisiones en la empresa barranquillera. *Universidad del Norte (Barranquilla, Colombia)*, 1-38.

CRISP-DM - Process Model. (s.d.). . Recuperado Agosto 27, 2010, a partir de http://www.crisp-dm.org/Process/index.htm

Danger, R., & Berlanga, R. (2001). Búsqueda de Reglas deAsociación en bases de datos ycolecciones de texto. Presented at the Departament de Llenguatges i Sistemes Informàtics, Universitat Jaume I.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases., 49.

Gaik Yee, C., Aziz, M. A. A., & Hasan, S. S. (2000). Applying Business Intelligence in Marketing Campaign Automation. *Second International Conference on Computer Research and Development.*

Gómez Vargas, A. M., & Castillo, M. (2005, Junio 15). Modelo de decisión para el proceso de mercadeo de nuevos vehículos en GM Colmotores.

Grahne, G., & Zhu, J. (2003). Efficiently Using Prefix-trees in Mining Frequent Itemsets. Recuperado a partir de http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.3.6241

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. New York, USA.

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. *DATA MINING AND KNOWLEDGE DISCOVERY, 8*, 53--87.

Holt, J. D., & Chung, S. M. (2001). Multipass Algorithms for Mining Association Rules in Text Databases. *Knowledge and Information Systems, 3*(2), 168-183. doi:10.1007/PL00011664

Hornick, M. F., Marcadé, E., & Venkayala, S. (2007). *Java Data Mining: Strategy, Standard, and Practice* (1st ed., pág 93). Morgan Kaufmann Publisher.

KNIME | Konstanz Information Miner. (s.d.). . Recuperado Enero 23, 2011, a partir de http://www.knime.org/

Kotler, P., Armstrong, G., Saunders, J., & Wong, V. (1999). *Principles of marketing* (2nd ed.). London: Prentice-Hall Europe.

Liu, Y., & Guan, Y. (2009). Application in Market Basket Analysis Based on FP-growth Algorithm. *World Congress on Computer Science and Information Engineering*.

Luna, J., Olmo, J., Romero, J., & Ventura, S. (2010). Minería de Reglas de Asociación con Programación Genética.

MOLINA, L. C., & RIBEIRO, S. (2010). Descubrimiento conocimiento para el mejoramiento bovino usando técnicas de data mining. *Actas del IV Congreso Catalán de Inteligencia Artificial.Barcelona*, 123-130.

NARANJO, R., & Sierra, L. (2009, Abril). Software tool for analysing the family shopping basket without candidate generation. *Ingeniería Investigativa*, *29*, 60-68.

Pacheco, C., & Ernst, M. D. (2005). Eclat: Automatic generation and classification of test inputs. *IN 19TH EUROPEAN CONFERENCE OBJECT-ORIENTED PROGRAMMING*, 504--527.

Pietracaprina, A., & Zandolin, D. (2003). Mining Frequent Itemsets using Patricia Tries. *In Proceedings of the Workshop on Frequent Itemset Mining Implementations, Melbourne, Florida, USA,*. Recuperado a partir de http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.437

Rapid - I. (s.d.). . Recuperado Enero 23, 2011, a partir de http://rapid-i.com/content/view/181/190/

SAP Andeancarib - SAP Professionals - ¿Qué son los módulos SAP? (s.d.). . Recuperado Enero 22, 2011, a partir de http://www.sap.com/andeancarib/ecosystem/sap_professionals/modules/index.epx

Sarawagi, S., Thomas, S., & Agrawal, R. (2000). Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications. *Data Mining and Knowledge Discovery*, *4*(2), 89-125.

Shenoy, P., Bhalotia, G., Haritsa, J. R., Bawa, M., Sudarshan, S., & Shah, D. (2000). Turbo-charging Vertical Mining of Large Databases. *IN PROC. OF ACM SIGMOD INT. CONF. ON MANAGEMENT OF DATA*, 22--33.

Shi, W. P., & Zhao Yu-lin, B. J. (2009). Mining Association Rules Based on Apriori Algorithm and Application. *International Forum on Computer Science-Technology and Applications*.

Software Contable Trident Enterprise. (s.d.). . Recuperado Agosto 20, 2010, a partir de http://www.basevirtual.com.co/index.php?option=com_content&view=article&id=110&Itemid=181

Sosa Sierra, M. del C. (2004, Noviembre 4). Inteligencia artificial en la gestión financiera empresarial. *Universidad del Norte (Barranquilla, Colombia)*, 153-186.

Timarán Pereira, R., Andrés, A., Ramírez, I., Alvarado, C., & Guevara, F. (s.d.). Análisis de desempeño de EquipAsso: Un algoritmo para el cálculo deItemsets frecuentes basado en operadores algebraicos relacionales. *Universidad de Nariño, Departamento de Ingeniería*.

Timarán, R., Calderón, A., Ramírez, I., Alvarado, C., & Guevara, F. (s.d.). Análisis de desempeño de EquipAsso: Un algoritmo para el cálculo de Itemsets frecuentes basado en operadores algebraicos relacionales. Universidad de Nariño, Departamento de Ingeniería.

Uno, T., Asai, T., Uchida, Y., & Arimura, H. (2003). LCM: An efficient algorithm for enumerating frequent closed item sets. *IN PROCEEDINGS OF WORKSHOP ON FREQUENT ITEMSET MINING IMPLEMENTATIONS (FIMI'03, 90.*

Recuperado a partir de

http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.2642

Weka 3 - Data Mining with Open Source Machine Learning Software in Java. (s.d.). . Recuperado Enero 23, 2011, a partir de http://www.cs.waikato.ac.nz/ml/weka/

Wu, J. (2010). Computational Intelligence-based Intelligent Business Intelligence System Concept and Framework. *Second International Conference on Computer and Network Technology.*

Xindo, W., & Vipin, K. (2009). *The top ten algoritms y data mining.* London,NewYork.