



*Universidad de las Ciencias Informáticas*

*Facultad 2*

***Título: Herramienta informática aplicando reglas  
de asociación en el proceso docente y productivo***

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

***Autores:***

*Rutminy Más Padilla*

*Yankiel Belén Suárez*

***Tutora: MSc. Eliana Bárbara Ril Valentin***

La Habana, Junio de 2013

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 2 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

**Firma de los Autores:**

Rutminy Más Padilla

---

Yankiel Belén Suárez

---

**Firma del Tutor:**

MSc. Eliana Bárbara Ril Valentin

---

## *Dedicatoria*

*A mi madre por ser, más que madre, mi amiga, mi hermana y mi tesoro más  
preciado.*

*A mi hermanita por ser la luz de mis ojos.*

*Rutminy*

*Dedicado a Tania Belén Suarez, María Cleofe Suarez Hernández, María  
Josefa Belén Suarez, Dora María Izquierdo y María Antonia Izquierdo.*

*YanKiel*

# *Agradecimientos*

## **Rutminy**

*A mi madre, que amo con la vida, por darme las fuerzas para lograr mis objetivos, su apoyo incondicional para hacerme saber que siempre estará a mi lado, su ejemplo para guiarme, por su sacrificio y dedicación para formarme en la vida. Gracias por ser mi razón de ser.*

*A Sheila y familia por su preocupación, apoyo y atención. Gracias por formar parte de mi vida.*

*A mi padrino por haber sido como un padre para mí.*

*A Isabel por ocupar un gran espacio en mi corazón.*

*A mi familia de amigos por su confianza y apoyo, a Mónica y a Yanet por ser mis amigas en todos los tiempos.*

*A mi compañero de tesis por su esfuerzo y empeño, sin él no hubiera sido posible la culminación de este trabajo.*

*A todos mis compañeros por compartir tantos momentos agradables, difíciles e inolvidables.*

*A mis amigos, especialmente a Lien, Maylen, Yinet, Mary, Saly, Liany, Osmar, Yary, Dany, Yasma y Roly por su cariño, apoyo, paciencia y confianza, por soportarme y por hacer de nuestra convivencia los mejores momentos de la universidad. Gracias por su amistad.*

*A Norey y Maylín por sacarme de tantos apuros en la tesis.*

*A todos los que, de una manera u otra, aportaron su granito de arena en mi desarrollo profesional.*

## **Yankiel**

*..... a todos los que de una forma u otra ayudaron a la realización de este sueño.*

## *Resumen*

Los avances tecnológicos en las últimas décadas han facilitado enormemente el acceso a grandes volúmenes de datos. Esta cantidad de información almacenada puede ser procesada para obtener mejoras en el funcionamiento de organizaciones e instituciones. Ejemplo de esto se evidencia en la Universidad de las Ciencias Informáticas donde existen varias plataformas para almacenar gran cantidad de información. Entre estas se encuentran Gestión Universitaria y Gestión de Proyectos que almacenan información docente y productiva de los estudiantes respectivamente, las cuales contribuyen a procesar dicha información para la toma de decisiones.

Con este fin se realiza la fundamentación teórica incluyendo conceptos y valoraciones que conforman la base teórica de la presente investigación. Para el desarrollo de la Herramienta Informática se utilizan métodos científicos de investigación, se realiza un estudio de los principales trabajos que vinculan la Minería de datos con el entorno educativo en la actualidad, así como las herramientas a usar; Python como lenguaje de programación, Eclipse como entorno de desarrollo y PostgreSQL como gestor de bases de datos. Además se utilizan las metodologías CRISP-DM como guía para el descubrimiento de conocimientos y XP para el desarrollo del software. Se obtiene como resultado una Herramienta Informática que emplea Reglas de asociación para apoyar el proceso docente y productivo teniendo en cuenta las asociaciones entre las notas de las asignaturas del departamento de Ingeniería de Software impartidas hasta el quinto semestre y las asociaciones entre estas notas y los roles que tienen asignados los estudiantes en un proyecto productivo.

### **PALABRAS CLAVE**

Descubrimiento de conocimiento, entorno educativo, Minería de datos.

# *Tabla de contenido*

Introducción .....	1
Capítulo 1. Fundamentación Teórica.....	8
1.1. Introducción.....	8
1.2. El proceso de descubrimiento de conocimiento en bases de datos.....	8
1.3. Minería de datos.....	10
1.3.1. Modelos de Minería de datos .....	10
1.3.2. Técnicas de Minería de datos.....	11
1.3.3. Reglas de asociación.....	11
1.4. Análisis crítico de trabajos similares en entorno educativo .....	14
1.5. Herramientas utilizadas en el proceso de Minería de datos.....	17
1.5.1. Pentaho .....	18
1.5.2. RapidMiner .....	18
1.5.3. WEKA vs RapidMiner .....	19
1.6. Metodologías.....	20
1.6.1. Metodologías para enfrentar un proceso de KDD .....	20
1.6.2. Metodologías de desarrollo de software .....	24
1.7. Lenguajes de programación.....	27
1.7.1. Python.....	27
1.7.2. PL/PgSQL.....	28
1.8. Entorno de Desarrollo Integrado .....	28
1.9. Interfaz gráfica de usuario.....	28
1.9.1. PyQt: Qt para Python.....	29

# *Tabla de contenido*

1.9.2. QtDesigner.....	29
1.10. Conclusiones del capítulo .....	29
Capítulo 2. Características de la herramienta informática propuesta.....	31
2.1. Introducción.....	31
2.2. Objeto de automatización.....	31
2.3. Solución propuesta.....	32
2.4. Fases de CRISP-DM.....	32
2.4.1. Comprensión del negocio .....	33
2.4.2. Comprensión de los datos .....	33
2.4.3. Preparación de los Datos .....	34
2.4.4. Modelado .....	35
2.4.5. Evaluación .....	35
2.4.6. Despliegue.....	36
2.5. Requerimientos no funcionales .....	36
2.6. Usuarios de la Herramienta Informática .....	38
2.7. Fase de exploración .....	38
2.7.1. Lista de funciones.....	39
2.7.2. Historias de Usuario .....	40
2.8. Fase de planeamiento.....	49
2.8.1. Estimación de esfuerzos.....	50
2.9. Fase de iteraciones .....	50
2.10. Conclusiones del capítulo .....	51
Capítulo 3. Diseño de la herramienta informática propuesta.....	52

# *Tabla de contenido*

3.1. Introducción.....	52
3.2. Propuesta de arquitectura de la Herramienta Informática.....	52
3.3. Patrones .....	53
3.3.1. Patrones de diseño GoF (Gang of Four) .....	53
3.3.2. Patrones para Asignar Responsabilidades (GRASP).....	55
3.4. Estructura del proyecto .....	57
3.5. Tarjetas Cargo o clase, Responsabilidad y Colaboración (CRC).....	57
3.6. Diseño de la base de datos .....	58
3.7. Conclusiones del capítulo .....	58
Capítulo 4. Implementación y Pruebas.....	59
4.1. Introducción.....	59
4.2. Fase Implementación .....	59
4.2.1. Tareas para cada HU.....	59
4.2.2. Iteración 1 .....	59
4.2.3. Iteración 2 .....	62
4.2.4. Iteración 3 .....	63
4.3. Fase Prueba .....	64
4.3.1. Pruebas unitarias .....	64
4.3.2. Pruebas de aceptación .....	65
4.4. Conclusiones del capítulo .....	66
Conclusiones Generales.....	67
Recomendaciones .....	68
Referencias bibliográficas.....	69

# Introducción

Los avances tecnológicos en las últimas décadas han facilitado enormemente el acceso a grandes volúmenes de datos. La necesidad de guardar información en sistemas informáticos; desde sus comienzos estuvo limitada, pero con el avance de la tecnología surgen las Bases de Datos (BD) permitiendo el exponencial incremento del volumen de información a almacenar. Esto fue suficiente para satisfacer a la mayoría de las necesidades diarias de las organizaciones, pero trajo el efecto secundario de la incapacidad para analizar y transformar esta gran cantidad de información en útil conocimiento para mejorar el funcionamiento de estas organizaciones. Bajo estas condiciones, surge el proceso de descubrimiento de conocimiento en Bases de Datos (KDD<sup>1</sup>)(Leyva 2011).

KDD apunta a procesar automáticamente grandes cantidades de datos para encontrar conocimiento útil en ellos, de esta manera permite al usuario el uso de esta información valiosa para su conveniencia. KDD es el “proceso no trivial de identificación de patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos” (Marín 2009).

El proceso KDD cuenta con cinco etapas donde se encuentra la Minería de datos (MD) (Lent 2010) que apareció en los años ´90. La MD o Data Mining<sup>2</sup> puede definirse como “una extracción de información desconocida no trivial y potencialmente útil de una gran cantidad de información”(Acosta 2007). Bajo este término se engloban un conjunto de técnicas de análisis cuyo objetivo es extraer conocimiento implícito de la base de datos.

Las bases de la Minería de datos se encuentran en la Inteligencia Artificial y en el análisis estadístico. Mediante los modelos extraídos utilizando técnicas de Minería de datos se aborda la solución a problemas de predicción, clasificación y segmentación. Un proceso típico de Minería de datos parte de la selección del conjunto de datos, tanto en lo que se refiere a las variables dependientes, como a las variables objetivo, como posiblemente al muestreo de los registros disponibles. Como consecuencia de este análisis, al conjunto de datos de entrada se le aplican una serie de

---

<sup>1</sup>Por sus siglas en inglés: Knowledge Discovery in Databases.

<sup>2</sup>En español, Minería de Datos (MD).

transformaciones con el objetivo de prepararlo para aplicar la técnica de Minería de datos que mejor se adapte a los datos y al problema. Finalmente se selecciona la técnica de minería, se construye el modelo predictivo o el descriptivo y se evalúan los resultados contrastando con un conjunto de datos previamente reservado para validar la generalidad del modelo (Cardona 2010).

En el caso de los modelos descriptivos se identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados. Por ejemplo, una agencia de viaje desea identificar grupos de personas con los mismos gustos, con el objeto de organizar diferentes ofertas para cada grupo y poder así remitirles esta información; para ello analiza los viajes que han realizado sus clientes e infiere un modelo descriptivo que caracteriza estos grupos (Bustamante 2010).

Existen innumerables ejemplos de aplicaciones de la Minería de datos en distintos sectores de la investigación, algunos de ellos (Ucan 2011):

- **Comercio y banca:** segmentación de clientes, previsión de ventas, análisis de riesgos.
- **Medicina y farmacia:** diagnóstico de enfermedades y la efectividad de los tratamientos.
- **Seguridad y detección de fraude:** reconocimiento facial, acceso a redes no permitidas.
- **Educación:** análisis del rendimiento académico en los estudios de informática de la Universidad Politécnica de Valencia aplicando técnicas de Minería de datos (Alcover 2007), Detección de Patrones de Producción Educativa Basada en Minería de datos (Molteni 2007), Sistemas de gestión de contenidos de aprendizaje y técnicas de Minería de datos para la enseñanza de ciencias computacionales: un caso de estudio en el norte de Coahuila (Juan Ramón Olague Sánchez 2010).

En el ámbito de la educación, la herramienta de la MD como técnica de análisis se ubica en el entorno del sistema educativo. Las bases de datos que se utilizan en los sistemas educativos permiten disponer de una gran cantidad de información, tanto de los estudiantes, trabajadores, departamentos, universidades, etc.(Fonseca 2011).

Una preocupación fundamental en la Educación Superior a nivel mundial surge de numerosos y desfavorables indicadores de deserción, desempeño y bajo rendimiento académico en alumnos de primer año de carreras universitarias. Esto ha llevado a las

universidades del mundo a investigar las causas que subyacen en esta problemática (Porcel 2010).

Cuba ha estado inmersa en un profundo y novedoso proceso de transformaciones educacionales y sociales como parte de los programas de la Batalla de Ideas. Con el fin de elevar el nivel cultural de la población y su calidad de vida se emprendieron nuevos programas destinados a satisfacer dichas premisas. En estas circunstancias surge la Universidad de Ciencias Informáticas con una misión fundamental: formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática así como producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la informática (Cabrera 2012).

En la Universidad de las Ciencias Informáticas (UCI) existe un gran volumen de información personal y docente digitalizada desde sus inicios en el 2003. Dicha información se encuentra actualmente almacenada en la plataforma de Gestión Universitaria, la cual ha superado las capacidades humanas para analizar y transformar la información en conocimiento útil que apoye la toma de decisiones. Es por ello, que factores como las notas de los estudiantes en una determinada disciplina curricular en el centro de Telemática (TLM) de la Facultad 2, no se toman en cuenta para la selección del rol más idóneo en los equipos de proyectos. Esto puede incidir en los estudiantes, la facultad, el proyecto y, en general, en la universidad, ya que no es solo importante satisfacer las necesidades del proyecto o la facultad sino saber en cuál rol se desempeñaría mejor el estudiante según sus conocimientos. Actualmente los roles son asignados en dependencia de las necesidades de dicho centro y hasta entonces no se han tenido en cuenta las notas de las asignaturas. Lo cual puede tener consecuencias tales como inconformidad y descontento para los estudiantes teniendo en cuenta que no tienen los conocimientos necesarios para enfrentar el rol asignado, así como atrasos para el proyecto ya que los estudiantes deben disponer de un tiempo para vencer dichos conocimientos.

Para enmendar lo antes planteado se requiere analizar cómo se comportan las asociaciones entre las notas de las asignaturas y los roles que desempeñan los estudiantes, con el objetivo de brindarles a los jefes de proyectos y de centro, de forma organizada, una información útil y efectiva para apoyar la toma de decisiones en la asignación de roles.

Teniendo en cuenta la **problemática** anteriormente descrita, en la UCI se hace necesario contar con métodos eficientes y automáticos, procesando de forma rápida y fiable la información para encontrar patrones de conocimiento apropiados para analizar la información docente y productiva de los estudiantes. Siendo dicha información las notas de los estudiantes en las asignaturas del departamento de ISW hasta el quinto semestre y el rol asignado en proyectos productivos.

De ahí que el **problema a resolver** sea: las insuficiencias en el descubrimiento de conocimiento en la información docente y productiva de los estudiantes de la UCI, están afectando la toma de decisiones en el proceso educativo. Teniéndose como **objeto de estudio** el descubrimiento de conocimiento en la información docente y productiva.

La investigación tiene como **objetivo general** desarrollar una herramienta informática aplicando Reglas de asociación para la obtención de patrones de comportamiento en la información docente de los estudiantes de la UCI que sirva de apoyo a la toma de decisiones en el proceso educativo y productivo, correspondiendo el **campo de acción** con aquellas herramientas informáticas para aplicar Reglas de asociación en la información docente y productiva. Para cumplir este objetivo general se trazaron los siguientes **objetivos específicos**:

1. Definir el marco teórico de la investigación identificando los conceptos y tendencias actuales relacionadas al Descubrimiento de conocimiento, Minería de datos, la tarea descriptiva Reglas de asociación y las soluciones informáticas más usadas para la obtención de patrones de comportamiento a partir de grandes fuentes de datos en entornos educativos.
2. Diseñar un modelo de solución de la Herramienta Informática propuesta que permita la realización de la tarea descriptiva Reglas de asociación y la realización de los procesos de Preparación de datos, Minería de datos y Difusión de resultados.
3. Implementar los módulos Preparación de datos, Minería de datos y Difusión de resultados cumpliendo con estándares de codificación y calidad definidos en la UCI.
4. Validar la herramienta informática propuesta a partir de pruebas de aceptación y unitarias a cada funcionalidad de la misma.

Para alcanzar los objetivos trazados y dar solución al problema planteado se elaboraron las siguientes **tareas de investigación**:

1. Análisis del proceso de descubrimiento de conocimientos en base de datos y los conceptos asociados a la Minería de datos con el propósito de conformar una base teórica para el desarrollo de la Herramienta Informática.
2. Revisión de los principales trabajos que vinculen la Minería de datos con el entorno educativo en la actualidad para definir la posición de los investigadores.
3. Selección de una metodología como guía para el proceso de descubrimiento de conocimiento a seguir en la implementación de la Herramienta Informática.
4. Selección de una metodología de desarrollo que sirva de guía en el proceso de desarrollo de la Herramienta Informática.
5. Selección de una biblioteca de Minería de datos para utilizar los algoritmos contenidos en la misma que respondan a la tarea descriptiva Reglas de asociación.
6. Definición de una arquitectura con el fin de obtener una visión general de la herramienta informática que se propone.
7. Identificación de requisitos fundamentales del negocio dentro del proceso de descubrimiento de conocimiento para su posterior implementación.
8. Definición de componentes reutilizables de las herramientas que permitan la realización de la tarea descriptiva Reglas de asociación.
9. Análisis de los estándares de codificación y calidad definidos en la UCI a ser utilizados para desarrollar la Herramienta Informática que se propone.
10. Realización de pruebas de funcionalidad a la Herramienta Informática para verificar el correcto funcionamiento de la misma.

Con el cumplimiento de dichas tareas se pretende obtener los siguientes **resultados**:

1. Definición de tres procesos (Preparación de datos, Minería de datos y Difusión de resultados) para realizar el Descubrimiento de conocimiento en una base de datos académica y productiva.
2. Herramienta informática que aplique Reglas de asociación en el proceso docente y productivo de los estudiantes de la UCI y que sirva de apoyo a la toma de decisiones en dicho proceso.

Se utilizaron **métodos científicos** que permitieron el correcto desarrollo de la investigación a partir de una caracterización del objeto de estudio y el estudio de las tendencias actuales sobre el contexto en el que se enmarca la problemática.

Métodos teóricos:

1. Analítico-sintético: este método fue utilizado en todo el proceso investigativo, ya que el análisis permitió descomponer todo el problema en varias partes para encontrar factores que posibilitaron una mejor comprensión del mismo y a su vez la síntesis permitió descubrir las relaciones que guardan entre sí estas partes como un todo. Por ejemplo, en la problemática a resolver es necesario hacer un análisis sobre el comportamiento de cada estudiante teniendo en cuenta las notas de las asignaturas del departamento de Ingeniería de Software y el rol que desempeña en un proyecto productivo, por separado, y después la síntesis permitirá descubrir las relaciones que existen entre estos factores. La presente investigación además sirvió para analizar las teorías, documentos, etc., permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.
2. Inductivo-deductivo: se utilizó dicho método para obtener una muestra representativa de los diferentes trabajos asociados a la Minería de datos en el entorno educativo que refleja lo que hay en común en los trabajos de forma individual y llevarlo a un conocimiento general, para después llevar el conocimiento a un caso particular que posibilite darle solución a la problemática anterior.
3. Histórico-lógico: fue de gran importancia para elaborar la fundamentación teórica de la investigación, ya que permitió estudiar lo más relevante en el plano teórico acerca del surgimiento de la Minería de datos, así como las metodologías, herramientas y tecnologías que se utilizarán para el desarrollo del trabajo.

4. Modelación: este método brindó la posibilidad de representar las propiedades y funcionalidades de la Herramienta Informática desarrollada llevándose a cabo el modelo teórico donde se representaron las características y relaciones fundamentales del proceso de KDD para una mejor comprensión en cada una de las etapas en que se desarrollará este trabajo.

Método empírico:

1. Observación: se puso de manifiesto durante todo el transcurso de la investigación con el objetivo de diagnosticar o explorar, como se realizaba en el centro TLM la selección del rol a los estudiantes que se incorporan en tercer año a un proyecto productivo.

Para abarcar el estudio realizado y la descripción de la solución propuesta el presente trabajo se compone de 4 capítulos. En el **capítulo 1** se realiza la fundamentación teórica que incluye conceptos y valoraciones que conforman la base teórica del tema tratado. Posteriormente en el **capítulo 2** se aborda la comprensión y planificación del negocio, así como el despliegue del proceso de descubrimiento de conocimientos para conformar la propuesta de solución según la problemática planteada. Luego para el **capítulo 3** se presenta de manera general la estructura de la Herramienta Informática a partir de las funcionalidades previstas y se realiza la implementación de la arquitectura como un todo, basándose en los resultados obtenidos en el diseño de la herramienta. Finalmente en el **capítulo 4** se realizan las tareas de ingeniería correspondientes a cada iteración y se valida la Herramienta Informática propuesta a partir de pruebas de aceptación y unitarias a cada funcionalidad de la misma.

# Capítulo 1. Fundamentación Teórica

## 1.1. Introducción

El presente capítulo aborda el proceso KDD, Minería de datos, tareas y algoritmos para obtener modelos o patrones a partir de datos. Se realiza un estudio orientado a dar solución a la problemática planteada, puntualizando las metodologías para el desarrollo de software así como las más utilizadas para guiar el proceso de KDD, tecnologías, herramientas, lenguajes y notaciones empleadas para tales propósitos. De igual manera se investigan diferentes trabajos relacionados con la Minería de datos en el entorno educativo para justificar el desarrollo de dicha Herramienta Informática.

## 1.2. El proceso de descubrimiento de conocimiento en bases de datos

KDD es el “proceso no trivial de identificación de patrones válidos, novedosos, potencialmente útiles y, en última instancia, comprensibles a partir de los datos” (Marín 2009). Este proceso completo se encarga de la preparación de los datos y de la interpretación de los resultados obtenidos. Se trata de interpretar grandes cantidades de datos y encontrar relaciones o patrones (López 2006).

Las principales etapas que componen el KDD se relacionan en la Figura 1.1 como se muestra a continuación:

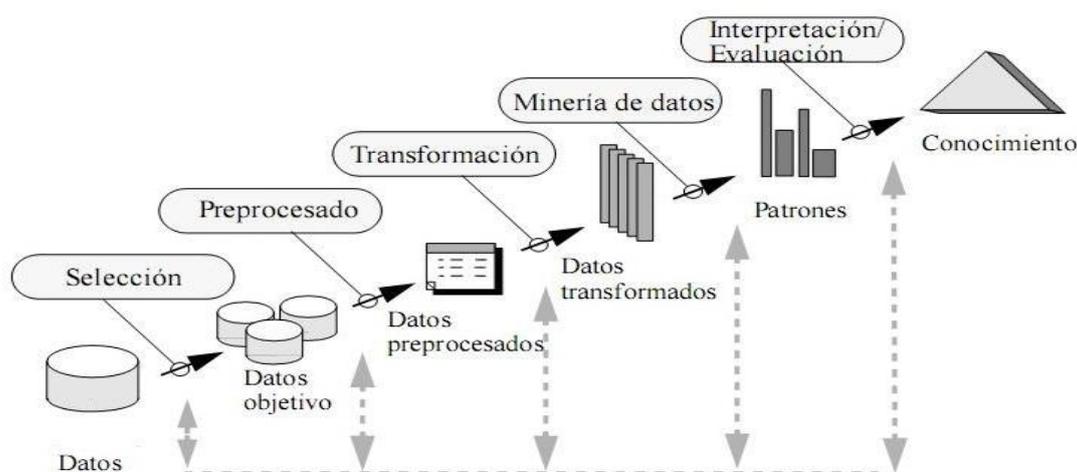


Figura 1.1. **Etapas del KDD** (Ril 2012)

## *Capítulo 1. Fundamentación Teórica*

Como se muestra en la figura anterior, el proceso KDD está distribuido en cinco fases (Consultores 2011) que se explican a continuación, además se puede observar cómo se enfocan en la Herramienta Informática propuesta.

- 1- Selección de datos. En esta etapa se determinan las fuentes de datos y el tipo de información a utilizar. Es la etapa donde los datos relevantes para el análisis son extraídos desde la o las fuentes de datos.
- 2- Pre procesado. Esta etapa consiste en la preparación y limpieza de los datos extraídos desde las distintas fuentes de datos en una forma manejable, necesaria para las fases posteriores.
- 3- Transformación. Consiste en el tratamiento preliminar de los datos, transformación y generación de nuevas variables a partir de las ya existentes con una estructura de datos apropiada.
- 4- Minería de datos. Es la fase de modelamiento propiamente tal, en donde métodos inteligentes son aplicados con el objetivo de extraer patrones previamente desconocidos, válidos, nuevos, potencialmente útiles y comprensibles y que están contenidos u “ocultos” en los datos.
- 5- Interpretación y Evaluación. Se identifican los patrones obtenidos y que son realmente interesantes, basándose en algunas medidas y se realiza una evaluación de los resultados obtenidos.

Teniendo en cuenta las etapas antes mencionadas se puede afirmar que el proceso KDD permite la limpieza de los datos, así como la selección, y transformación, con el alcance de extraer patrones de estos datos para convertirlos en conocimiento.

Se puede resumir que el KDD tiene como objetivo principal encontrar conocimiento útil, válido, relevante y nuevo sobre un fenómeno o actividad mediante algoritmos eficientes, dadas las crecientes órdenes de magnitud de los datos (Marín 2009; Marín 2009).

Para la realización de la Herramienta Informática propuesta se necesita llevar a cabo el proceso completo de KDD para lograr cumplir los objetivos planteados en la presente investigación.

## 1.3. Minería de datos

Dentro de las etapas del proceso de KDD se encuentra la Minería de datos, siendo esta la etapa más distintiva dentro de dicho proceso.

Existen diversas definiciones para la Minería de datos aportadas por diferentes autores, pero en su esencia, coinciden en que la tarea fundamental es descubrir conocimiento (reglas, patrones) a partir de grandes volúmenes de datos, apoyados en técnicas o herramientas (automáticas o asistidas), de tal manera que su uso ayude a tomar decisiones más seguras que reporten algún tipo de beneficio a las organizaciones (Sarasa 2008).

El objetivo de la Minería de datos es extraer información interesante, útil a partir de datos. En muchos casos esta información tiene la forma de patrones no evidentes en los datos y, dependiendo de su complejidad, puede llegarse a hablar de conocimiento (González 2012).

La MD está basada en técnicas de la estadística y el aprendizaje en máquina, además se relaciona estrechamente con Bases de Datos y brinda soluciones a problemas prácticos obteniendo patrones entre la información guardada en las BD convirtiéndolos en conocimiento.

### 1.3.1. Modelos de Minería de datos

La Minería de datos pretende analizar datos para extraer conocimientos de estos, los cuales pueden presentarse en forma de relaciones, patrones o reglas que se infieren de los datos y que resultan novedosos, o resultan ser una descripción más precisa de los mismos.

El resultado obtenido constituye un modelo de los datos analizados, los cuales se pueden representar de diferentes formas y en dependencia de estas se determina un tipo de técnica que permite inferirlos.

Los modelos pueden clasificarse en dos tipos: predictivos y descriptivos (Larose 2009).

Los modelos predictivos pretenden estimar valores futuros o desconocidos de variables de interés, usando otras variables o campos de la base de datos, a las que nos referiremos como variables independientes o predictivas.

# Capítulo 1. *Fundamentación Teórica*

Los modelos descriptivos, identifican patrones que explican o resumen los datos, es decir, sirven para explorar las propiedades de los datos examinados, no para predecir nuevos datos, siendo este el modelo a seguir para darle solución a la problemática.

## **1.3.2. Técnicas de Minería de datos**

Las técnicas de Minería de datos intentan obtener patrones o modelos a partir de los datos recopilados. Decidir si los modelos obtenidos son útiles o no, suele requerir una valoración subjetiva por parte del usuario. Las técnicas de Minería de datos se clasifican en dos grandes categorías: supervisadas o predictivas y no supervisadas o de descubrimiento del conocimiento (Tuya 2007).

Los algoritmos supervisados o predictivos predicen el valor de un atributo (etiqueta) de un conjunto de datos, a partir de estos datos cuya etiqueta es conocida se induce una relación entre dicha etiqueta y otra serie de atributos para realizar la predicción (Quintales 2013).

Los métodos no supervisados o de descubrimiento del conocimiento descubren patrones y tendencias en los datos actuales, lo cual sirve para llevar a cabo acciones y obtener un beneficio de estas (Quintales 2013).

El proceso de Minería de datos requiere de establecer previamente los objetivos para el análisis de los datos que se disponen, en virtud de estos existen las llamadas tareas de minería, entre las cuales se decidió utilizar Reglas de asociación en la presente investigación para apoyar la toma de decisiones en el proceso docente y productivo. Las mismas se abordan en el siguiente epígrafe.

## **1.3.3. Reglas de asociación**

Las Reglas de asociación son una tarea descriptiva, la cual es el eje fundamental de esta investigación. Tiene como objetivo encontrar relaciones no explícitas entre atributos nominales y es empleada para reconocer cómo la ocurrencia de determinados sucesos puede provocar la aparición de otros (Despaigne 2011).

Las Reglas de asociación (RA) son aquellas reglas, formadas por un lado izquierdo al que se le llamará implicativo y un lado derecho llamado implicado, cada lado de la regla está formado por un conjunto de atributos. Analizando profundamente el concepto de RA y su significado, se tiene que, como se dijo anteriormente, una RA es

## Capítulo 1. Fundamentación Teórica

una relación de la forma  $A \rightarrow C$  ( $s$ ,  $c$ ) donde  $s$  representa el soporte y  $c$  la confianza. Un valor elevado de confianza indica que hay una gran probabilidad de que la aparición de  $A$  implique también la aparición de  $C$  y además si el valor del soporte es grande entonces esto quiere decir que  $A$  y  $C$  son elementos muy abundantes, por tanto influyentes a la hora de caracterizar los elementos que forman la BD que se analiza. Las RA se basan en medidas de confianza y soporte, consideran cualquier conjunto de atributos con cualquier otro conjunto de atributos; funcionan con atributos discretos (Limonta 2007).

La elegancia de expresar de forma clara y compacta los conocimientos descubiertos ha convertido a las RA en una herramienta poderosa. Tanto es así, que en los últimos años, con el auge de la minería de textos, se ha intentado aplicar conceptos similares para expresar patrones descubiertos en colecciones de documentos, pero este trabajo se centra en la extracción de las Reglas de asociación representativas solamente en base de datos.

Es necesario tener bien presente la dificultad que puede presentar la utilización de las RA, pues pueden generar un gran número de posibles reglas a considerar, incluso para conjuntos de datos pequeños, debido a que puede contener cualquier cantidad de combinaciones de pares atributo-valor (Alonso 2013). También tener en cuenta las desventajas que presentan: los valores adecuados para los parámetros son difíciles de establecer *a priori*, los resultados obtenidos dependen del orden de presentación de los patrones. Además, la extracción con RA presenta un inconveniente principal, estriba en que los costes tanto en tiempo como en espacio pueden ser inviables en bases de datos lo suficientemente voluminosas (Silvana 2009).

A pesar de estas dificultades el proceso de generación de estas reglas es bastante sencillo, aunque el principal problema es el gran número de reglas generadas donde muchas de estas no tienen utilidad, para lo cual es necesario evaluar su validez considerando los factores de soporte y de confianza.

Usar RA tiene ventajas sobre otras técnicas como la Clasificación, ya que requieren un menor número de atributos y brindan una mayor eficiencia, por lo cual puede ajustarse a un amplio conjunto de parámetros y usar cálculos sencillos recorriendo el conjunto de datos sólo una vez. Aunque una de las desventajas es que los resultados obtenidos dependen del orden de presentación de los patrones.

Para dar solución a la problemática antes expuesta se seleccionan las Reglas de asociación para describir el comportamiento de un conjunto de atributos, relacionando en este caso los resultados académicos de los estudiantes en las asignaturas pertenecientes al departamento de Ingeniería de Software en el transcurso de su carrera hasta el quinto semestre así como el rol que desempeña en un proyecto productivo.

### 1.3.3.1. Algoritmos de las Reglas de asociación

La forma de implementar cada una de las tareas de minería la constituyen los algoritmos. De la correcta elección de ellos dependerá la representación del conocimiento obtenido del análisis realizado a los datos ya sea en forma de regla, patrones o resumen (Limonta 2007).

Entre los algoritmos para esta técnica se pueden encontrar:

- A priori.
- Partition.
- Eclat.
- FP-Growth.

Para la selección del algoritmo a aplicar se realizaron varias comparaciones entre estos y se tomaron como base los siguientes criterios:

- Cantidad de consultas al conjunto de datos iniciales.
- Rendimiento.
- Implementación en RapidMiner.

En el caso del algoritmo Apriori realiza un número elevado de consultas al conjunto inicial de datos lo que representa una desventaja para la solución por la ralentización del proceso, al igual que es muy costosa para conjuntos de datos de grandes dimensiones. Por otro lado los algoritmos Partition y Eclat presentan un costo computacional elevado y no son implementados por RapidMiner; la cual es una herramienta que se propone utilizar para el descubrimiento de conocimiento y Minería de datos como se muestra en el epígrafe 1.5.2. Por último tenemos el FP-Growth que es implementado por RapidMiner, realiza menos consultas al conjunto de datos iniciales que el resto de los algoritmos analizados y para generar los ítems frecuentes

## Capítulo 1. Fundamentación Teórica

utiliza la estructura FP-Tree (árbol de patrones frecuentes), que para su creación no necesita de un costo computacional elevado.

Luego de realizadas las comparaciones se determinó utilizar el algoritmo FP-Growth, pues según las valoraciones resultantes fue el que más se ajustó a las necesidades de la solución.

El algoritmo FP-Growth además fue concebido con el objetivo de solucionar las fragilidades del algoritmo A priori. Consiste en el proceso de obtención de todos los patrones frecuentes de un conjunto de transacciones, mediante la generación del árbol FP que comprime las transacciones y el podado recursivo del mismo.

Este algoritmo en un primer paso borra todos los ítems del conjunto de datos iniciales que no son frecuentes individualmente o no sobrepasan en el mínimo soporte. Luego de este paso se genera el árbol FP-Tree. Un árbol FP-Tree es básicamente una estructura formada por los prefijos para las transacciones, cada rama del árbol representa el grupo de transacciones que comparten el mismo prefijo y cada nodo corresponde a un ítem. Todos los nodos que referencian al mismo ítem son referenciados juntos en una lista, de modo que todas las transacciones que contienen un ítem específico pueden encontrarse fácilmente y contarse al atravesar la lista. Esta lista puede ser acezada a través de la cabeza, lo cual también expone el número total de ocurrencias del ítem en la base de datos (Villavicencio 2012).

El FP-Growth realiza un recorrido en profundidad para la búsqueda de ítems frecuentes en el FP-Tree, este árbol está ordenado de forma decreciente comparando el contador de referencias de cada ítem obteniendo los de menor frecuencia en los niveles cercanos a la raíz, por lo que al recorrer en profundidad el primer ítem a visitar es el de mayor referencia. Este algoritmo tiene gran ventaja sobre el Apriori teniendo en cuenta la búsqueda por conjunto de ítems frecuentes ya que realiza la búsqueda solamente en el FP-Tree y no en todo el conjunto de datos (Pinho 2010).

### **1.4. Análisis crítico de trabajos similares en entorno educativo**

En el ámbito internacional, existen trabajos donde se hace énfasis en la Minería de datos educacional. El artículo "Una lectura sobre deserción universitaria en estudiantes de pregrado desde la perspectiva de la minería de datos"(Timarán 2010), describe los

## Capítulo 1. Fundamentación Teórica

objetivos, estrategias y resultados de un proceso de descubrimiento de conocimiento que se llevó a cabo en la Universidad de Nariño (Colombia) para determinar en la comunidad universitaria perfiles de bajo rendimiento académico y deserción estudiantil, para lo cual se utilizó la base de datos histórica de los estudiantes de pregrado. Se utilizó para generar reglas de asociación el algoritmo EquipAsso así como C4.5v para reglas de clasificación disponibles ambos implementados en la herramienta TaryKDD seleccionada para realizar este proceso. Teniendo en cuenta que:

- En la herramienta TaryKDD, el algoritmo Equipasso, aunque presenta mejores tiempos de ejecución, consume más recursos del sistema, en especial la memoria principal (Timarán 2013).
- Las estrategias seleccionadas para enfrentar la fase de preparación de los datos del KDD no se consideran óptimas al introducir datos ficticios donde existían campos vacíos en la fuente de datos.
- Las variables consideradas relevantes, ejemplo: si el colegio es público o privado, el tipo de calendario A o B y la jornada de estudio, seleccionadas para realizar el proceso de obtención de reglas de asociación en dicha universidad, no se consideran necesarias en el enfoque de la problemática presente en la UCI; por tanto se llega a la conclusión que la investigación no proporciona la solución satisfactoria.

“Usando Minería de datos para la continua mejora de cursos de E-Learning”(Ventura 2013). Este artículo presenta como objetivo general desarrollar una metodología cíclica para la mejora continua de cursos de e-learning, utilizando técnicas de minería de datos aplicadas a la educación. Con este fin se desarrolla una herramienta llamada CIECoM que descubre reglas de asociación entre los datos utilizados por los alumnos de un curso. A diferencia de la mayoría de las aproximaciones de minería aplicadas a la educación y que están orientadas al alumno, este caso se enfoca en ayudar al profesor a través de recomendaciones, para mejorar la estructura y los contenidos de un curso e-learning. En este caso las variables tratadas para la obtención de reglas de asociación (Nota media final del curso (c\_score), Número de mensajes leídos en el foro (c\_forum\_read), Tiempo total empleado en el cuestionario (c\_quiz\_time ),etc.) no se consideran necesarias para la problemática existente en la UCI. Para la utilización de la herramienta CIECoM se requiere que el usuario sea en cierto grado experto para que encuentre el balance entre los parámetros utilizados por el algoritmo seleccionado

## *Capítulo 1. Fundamentación Teórica*

para realizar el proceso de minería. Teniendo en cuenta los objetivos de la presente investigación y de acuerdo con los datos seleccionados a minar para encontrar asociaciones entre estos, este artículo se toma solo de referencia.

En el ámbito nacional se pueden encontrar trabajos relacionados con el entorno educacional.

El trabajo "Gestión del conocimiento para la obtención de patrones que contribuyan a la formación de equipos de proyectos de software"(Montalvo 2013), tiene como objetivo mostrar los resultados obtenidos a partir de la aplicación del proceso KDD a la muestra obtenida como resultado de un conjunto de test psicológicos, a un grupo numeroso de miembros de equipos de desarrollo pertenecientes a la Industria de software, con vistas a identificar patrones que contribuyan a la formación de un equipo de proyecto con posibilidades de lograr sinergia. En el desarrollo de este trabajo se utilizan las técnicas tanto del modelo predictivo (Selección de atributo y Clasificación), como del descriptivo (Reglas de asociación y clúster) de la Minería de datos. Si bien este trabajo aporta conocimiento para la presente investigación en cuanto a técnicas psicológicas para la conformación de equipos de proyectos más productivos, competitivos y comprometidos, solo aporta una metodología según las características psicológicas que deben tener las personas para ser asignados a los principales roles en el proceso de desarrollo de un proyecto. Además, todo el proceso se describe sobre el ámbito psicológico, obviando el desempeño docente en asignaturas indispensables como la Ingeniería de Software, siendo este un indicador importante para medir el conocimiento adquirido por el personal durante su preparación.

En el trabajo "Minería de datos: Herramienta de apoyo en la selección de equipos de proyectos informáticos"(Wilfred 2013), se expone el estudio realizado con el propósito de valorar la aplicabilidad de la minería de datos para el reconocimiento y extracción de información desconocida, que apoye la toma de decisiones vinculadas al proceso de adquisición o selección, dentro del área de la gestión de los recursos humanos de proyectos informáticos. Este trabajo valora dónde pueden ser más aplicables las técnicas de minería de datos en grupos de proyectos de software ya concluidos para clasificar nuevos proyectos de acuerdo con los grupos semejantes previamente identificados. Para realizar este proceso, se seleccionan diferentes técnicas de minería de datos para cada fase del proceso definido. Las reglas de asociación se encuentran seleccionadas entre las técnicas de minería para determinar grupos de ingenieros afines, considerando resultados individuales y colectivos alcanzados en proyectos

## *Capítulo 1. Fundamentación Teórica*

realizados, evaluaciones, encuestas o test de afinidad. Más que una herramienta, el trabajo referenciado propone un proceso con diferentes fases. Para cada fase se selecciona una o varias técnicas de minería según la fase correspondiente. La selección en ese trabajo de las Reglas de asociación aumenta la confianza en esta técnica para su utilización efectiva en la búsqueda de patrones de comportamiento. Por tanto, si bien no soluciona la problemática existente en la UCI, apoya la decisión de implementar la herramienta informática propuesta utilizando las reglas de asociación como técnica fiable para apoyar la toma de decisiones en esta institución.

Existen también trabajos desarrollados en la UCI donde no se puede dejar de mencionar la Tesis de Diploma “Obtención de Reglas y Patrones en el Proceso Académico de la Universidad de Ciencias informáticas”. Este tenía como objetivo principal predecir los resultados mediante el uso de las técnicas de agrupamiento y asociación para encontrar patrones entre los resultados académicos del primer año y la vinculación al origen social de estos estudiantes (Pérez 2007). Una vez analizada la tesis anterior se encuentran propósitos similares al problema a resolver, aunque esta usa datos muy generales pues recogen los resultados académicos de todas las asignaturas de primer año, además de vincular a los estudiantes a su origen social. Para la fase de minería se utiliza la herramienta privativa SQL Server 2005 implicando gastos para la UCI al obtener la licencia para su utilización. A diferencia de esta tesis, en el problemática de la presente investigación solo se necesitan describir los resultados en las asignaturas del departamento de ISW hasta el quinto semestre y el rol que ocupa en un proyecto productivo. Por tanto, se considera referenciar dicha tesis como apoyo en el desarrollo del trabajo.

### **1.5. Herramientas utilizadas en el proceso de Minería de datos**

A continuación se exponen las herramientas necesarias para llevar a cabo el proceso de descubrimiento de conocimientos. Teniendo en cuenta las ventajas que proveen las tecnologías de software libre sobre las privativas en cuanto a precio, accesibilidad, calidad y diseño se decide desarrollar este proceso sobre tecnologías libres atendiendo a la flexibilidad y características que poseen estas herramientas. Dentro de las herramientas más utilizadas y disponibles en el mercado para enfrentar un proceso KDD se pueden encontrar Pentaho, RapidMiner y Weka.

## 1.5.1. Pentaho

La compañía Pentaho es una alternativa de código abierto para la Inteligencia de Negocio (BI<sup>3</sup>), el cual desarrolla varias herramientas, una de ellas engloba a las demás, la cual se denomina Pentaho BI Suite Enterprise Edition que provee reportes, OLAP<sup>4</sup> (On-Line Analytical Processing), integración de datos, Minería de datos y una plataforma de BI. También representa una solución completa de herramientas para la integración de datos (Pentaho 2010).

Esta herramienta multiplataforma provee una alta gama de operadores para realizar selección, filtrado así como transformaciones a partir de datos extraídos de una fuente de datos. Pentaho posee un entorno gráfico de desarrollo apoyado en el uso de tecnologías Java, XML y

JavaScript incluyendo además cuatro herramientas para ejecutar transformaciones y trabajos; CHEF: para crear trabajos, Kitchen: para ejecutarlos; Spoon: para diseñar transformaciones ETL usando el entorno gráfico y PAN: para ejecutar transformaciones diseñadas con Spoon, esta última de gran utilidad para ejecutar la transformación necesaria de los datos extraídos, creada a partir de las tecnologías XML y JavaScript que soporta la herramienta.

Se decide utilizar Pentaho para la fase de Preparación de Datos.

## 1.5.2. RapidMiner

RapidMiner (Larman 2008) es una herramienta creada en la Universidad de Dortmund para el descubrimiento del conocimiento y la Minería de datos. Este entorno utiliza diferentes algoritmos de aprendizaje y otras utilidades, además de estar desarrollada en el lenguaje Java y funcionar en las herramientas operativas más conocidos, siendo de esta forma un software de libre distribución y de código abierto.

Se retroalimenta de las librerías de funciones de WEKA (Marcé 2011) en su entorno de aprendizaje, posee alrededor de 400 operadores que pueden ser combinados, usa el lenguaje de *scripting* XML<sup>5</sup> para describir los operadores y su configuración y puede ejecutarse por línea de comandos.

---

<sup>3</sup>Business Intelligence

<sup>4</sup> En español: Procesamiento Analítico en línea.

<sup>5</sup> del inglés: Extensible Markup Language

Para la realización de este trabajo resulta de gran utilidad el manejo de dicha herramienta para aplicar los algoritmos de minería, donde se espera obtener resultados con la calidad requerida.

### 1.5.3. WEKA vs RapidMiner

El paquete Weka (Waikato Environment for Knowledge Analysis) contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades. La versión original de Weka fue como un inicio para modelar algoritmos implementados en otros lenguajes de programación, más unas utilidades para el procesamiento de datos desarrolladas en C para hacer experimentos de aprendizaje automático. Esta versión original se diseñó inicialmente como herramienta para analizar datos procedentes del dominio de la agricultura, pero la versión más reciente basada en Java (WEKA 3), que empezó a desarrollarse en 1997, se utiliza en muchas y muy diferentes áreas, en particular con finalidades docentes y de investigación (Benavides 2007).

Los puntos existentes entre estas dos herramientas se listan a continuación:

- Disponibles libremente bajo la licencia pública general de GNU.
- Están implementados en Java.
- Multiplataforma.
- Ambos contienen colecciones de técnicas para preprocesamiento de datos y modelado.
- Diversas fuentes de datos (American Standard Code (ASCII), Java Database Connectivity (JDBC)).

Se decidió utilizar RapidMiner por algunas características específicas. Una de ellas es la utilización del lenguaje *scripting XML* para describir sus operadores así como las configuraciones de los mismos, siendo esta una ventaja a tener en cuenta dada la flexibilidad de diseño de un proceso. RapidMiner dispone de módulos para integrar otras herramientas de minería a su marco de trabajo, entre ellas Weka, utilizando sus operadores descritos con *scripting XML*.

## 1.6. Metodologías

Una metodología puede definirse como el estudio o elección de un método pertinente para un determinado objetivo (ISPJAE 2008). En este epígrafe se muestran las metodologías seleccionadas para enfrentar un proceso de KDD y para el desarrollo de la Herramienta Informática propuesta.

### 1.6.1. Metodologías para enfrentar un proceso de KDD

Actualmente en el mercado existen tres metodologías importantes para llevar a cabo proyectos de explotación de la información, entre las cuales se pueden encontrar (Moine 2010):

- SEMMA
- P<sup>3</sup>TQ
- CRISP-DM

#### 1.6.1.1. Metodología SEMMA

El nombre de dicha metodología hace referencia por sus iniciales a las cinco fases que se consideran al utilizarla: Muestrear (Sample), Explorar (Explore), Modificar (Modify), Modelar (Model) y Valorar (Assess). Esta es la primera metodología y fue propuesta por el SAS Institute, acrónimo de *statistical analysis systems*, especialmente para trabajar con el software de Minería de datos. Esta se enfoca en los aspectos técnicos y excluye las actividades de análisis y comprensión del problema en cuestión, además organiza sus herramientas para cada fase que compone dicha metodología.

En la Figura 1.2 se muestra la dinámica de la metodología SEMMA.

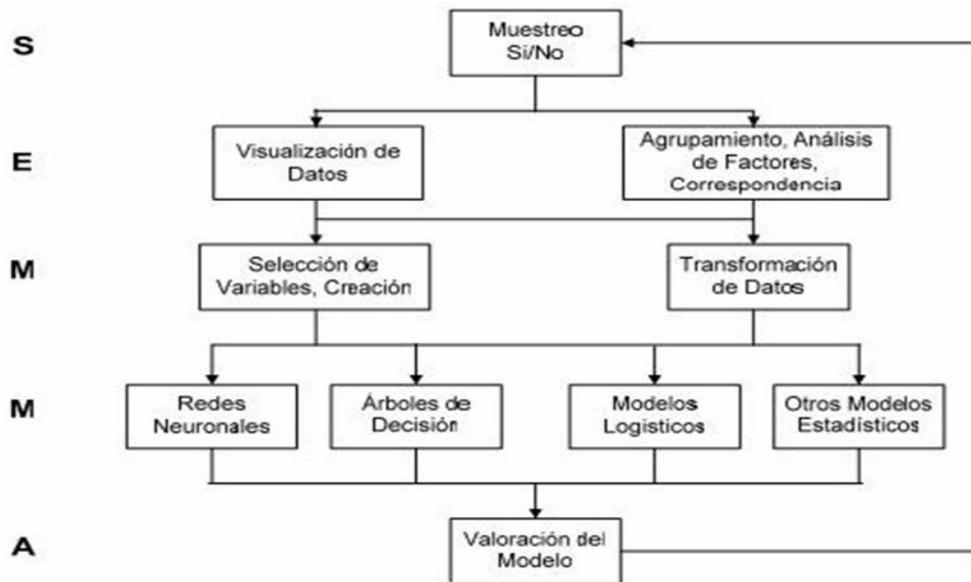


Figura 1.2. Metodología SEMMA (Espinosa 2010)

## 1.6.1.2. Metodología P<sup>3</sup>TQ

Su denominación original es Catalyst, pero fue difundida con este nombre por los términos Product, Place, Price, Time, Quantity (en español; Producto, Lugar, Precio, Tiempo y Cantidad). Esta metodología se divide en dos etapas, la primera Modelado de Negocio o MII, y la segunda Minería de datos, Modelo de Explotación de Información o MIII.

El Modelo de Negocio (MII), proporciona una guía de pasos para identificar un problema de negocio y los requerimientos reales de la organización.

El Modelo de Explotación de Información (MIII), cuenta con una guía de pasos para la construcción y ejecución de modelos de Minería de datos, construcción y ejecución de modelos de Minería de datos a partir del Modelo de Negocio (MII).

Más bien esta metodología es utilizada en proyectos que no tengan definido el problema del negocio haciéndose necesario analizar las relaciones precio, lugar, producto, tiempo y cantidad importantes para la empresa.

## 1.6.1.3. Metodología CRISP-DM

CRISP-DM (CRoss Industry Standard Process for Data Mining: Procedimiento Industrial Estándar para realizar Minería de datos). Esta metodología fue concebida de forma tal que resulte independiente de la herramienta que se utilice para el desarrollo

## Capítulo 1. Fundamentación Teórica

de un proyecto; y es de distribución libre, por lo que se encuentra en constante desarrollo por la comunidad internacional. CRISP-DM fue propuesta inicialmente por un consorcio de empresas encabezadas por Statistical Product and Service Solutions (SPSS) (Amil 2010) en 1996, y luego liberada para su empleo y desarrollo por parte de la comunidad internacional.

Esta metodología cuenta con seis etapas o fases fundamentales como guía durante el ciclo de vida de un proyecto que implemente un proceso de KDD, tal como se muestra en la Figura 1.3:

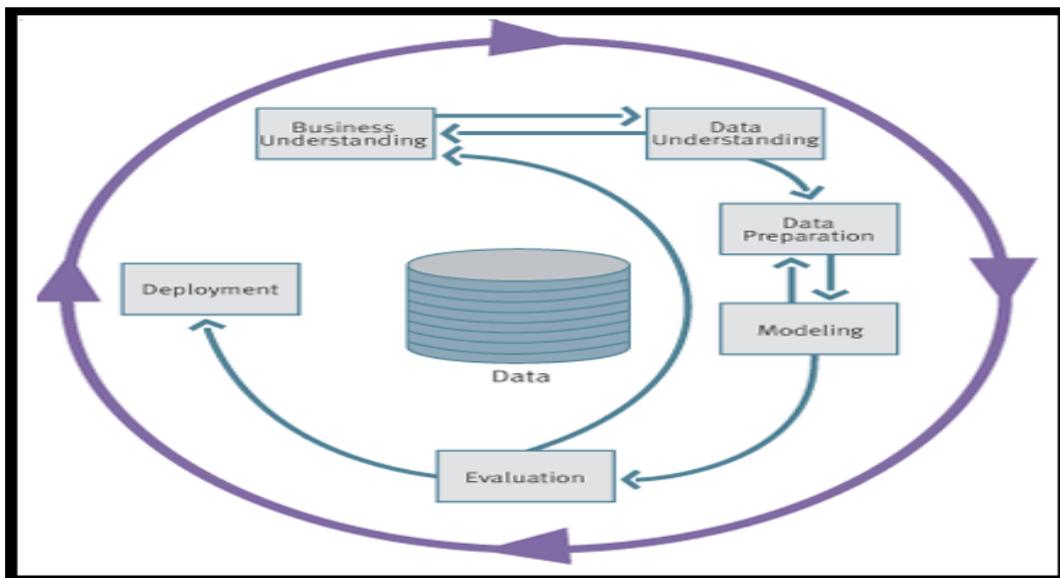


Figura 1.3. Fases de CRISP-DM (Matías 2010)

Según se explica en (Chapman 2012), las fases son detalladas brevemente a continuación:

- 1. Comprensión del Negocio:** se determinan los objetivos y requerimientos del negocio, se evalúa la situación para definir el problema como un proyecto de Minería de datos y se elabora el plan del proyecto.  
Enfocándose en dicha problemática primeramente se obtienen los datos necesarios de la plataforma Gestión Universitaria y Gestión de Proyectos donde se encuentra almacenada gran cantidad de información personal y docente de los estudiantes de la UCI así como sus roles en la producción.
- 2. Comprensión de los Datos:** se recopilan los datos iniciales, se describen, se verifica la calidad de los mismos y se exploran para detectar los primeros puntos de vista o subconjuntos de carácter interesante que permitan crear hipótesis sobre la información oculta.

## Capítulo 1. Fundamentación Teórica

En esta fase se seleccionaron los resultados de los estudiantes en las asignaturas pertenecientes al departamento de ISW hasta el quinto semestre y el rol que desempeñan en un proyecto productivo. Se pretende asociar dichas variables para describir el comportamiento de estos patrones.

3. **Preparación de los Datos:** se seleccionan los datos, se construyen, integran y estructuran para ser minados. Estas tareas incluyen tablas, registros, selección de atributos, así como la transformación y limpieza de los datos para que sean utilizados en las herramientas de modelado.

En esta fase se realiza la transformación de los datos, en este caso se transforman las notas de las asignaturas y se filtran las asignaturas y los roles que se pretendan asociar.

4. **Modelado:** se seleccionan y aplican las técnicas de modelado, se genera el diseño del experimento y se construyen y evalúan los modelos.

Se selecciona la técnica descriptiva Reglas de asociación para realizar las asociaciones entre los patrones definidos y se aplica el algoritmo FP-Growth.

5. **Evaluación:** se evalúa a fondo el modelo revisando el proceso para construirlo, se asegura que el modelo cumple apropiadamente con los objetivos del negocio, se determinan temas relevantes en el negocio que no hayan sido suficientemente considerados y se valoran los resultados.

6. **Despliegue:** se traza la estrategia de empleo de los resultados, se planifica el mantenimiento del proceso, se organiza y presenta el conocimiento obtenido como resultado de manera que el cliente pueda usarlo, y se documentan las experiencias.

Luego de las investigaciones iniciales, se concluye que SEMMA se centra en los aspectos técnicos de los proyectos de explotación de datos, está acotada ya que ha sido diseñada para ser implementada con los productos SAS<sup>6</sup> y se inicia analizando los datos. Mientras que CRISP-DM es más completa y abierta que SEMMA, es gratuito, se inicializa analizando los objetivos del negocio profundizando en mayor detalle sobre las tareas y actividades a ejecutar en cada etapa del proceso de Minería de datos orientándose a los objetivos empresariales y a una metodología de gestión de proyectos, por lo cual se utiliza como metodología para enfrentar un proceso de KDD en el desarrollo del presente trabajo.

---

<sup>6</sup>Compañía líder de software de Business Analytics y servicios y el mayor proveedor independiente del mercado de BI.

# Capítulo 1. Fundamentación Teórica

Es importante resaltar que esta metodología ofrece a las organizaciones e instituciones la estructura necesaria para obtener mejores resultados en la MD y tiene como ventaja que es una guía para inexpertos en la MD y es adaptable a las necesidades de una empresa o sector particular.

## 1.6.2. Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos informáticos. Habitualmente se utiliza el término “método” para referirse a técnicas, notaciones y guías asociadas, que son aplicables a actividades del proceso de desarrollo (Suárez 2009). Un proceso define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo (Muñiz 2012). No existe una metodología de software universal pues es necesario tener en cuenta las características específicas que exige cada proyecto por lo cual existen diferentes metodologías las cuales se pueden clasificar en tradicionales o ágiles.

### 1.6.2.1. Metodología Tradicional

Las metodologías tradicionales o pesadas se caracterizan por hacer mayor énfasis en la planificación y control del proyecto, así como en la precisa especificación de los requisitos y modelado. Estas metodologías se enfocan en conseguir un software más eficiente imponiendo una disciplina de trabajo sobre el proceso de desarrollo de software. Además, se centran en el control del proceso definiendo los roles, actividades, artefactos, herramientas y una documentación detallada. Una desventaja fundamental es que estas no se adaptan a los cambios en la definición de requisitos por lo cual no es adecuado utilizarlas en proyectos donde los requisitos puedan variar (EcuRed 2013).

### 1.6.2.2. Metodología Ágil

Las metodologías ágiles se centran en dar mayor valor a la interacción entre el cliente y el equipo de desarrollo estableciendo una estrecha comunicación entre estos, así como al desarrollo incremental del software con pequeñas iteraciones. Es efectivo usar el método ágil para proyectos donde cambien los requisitos o además se exija reducir los tiempos de desarrollo manteniendo la alta calidad del producto. Principalmente estas metodologías se enfocan en los siguientes valores (Deseta 2013):

1. Al individuo y las interacciones en el equipo de desarrollo más que a las actividades y las herramientas.
2. Desarrollar software que funciona más que conseguir una buena documentación, implica minimalismo respecto del modelado y la documentación del sistema.
3. La colaboración con el cliente más que la negociación de un contrato.
4. Responder a los cambios más que seguir estrictamente una planificación.

### 1.6.2.3. Metodología Tradicional v/s Ágil

Entre las metodologías tradicionales y ágiles existen grandes diferencias que son necesarias tener en cuenta para la selección de la metodología adecuada para el desarrollo del software. Las diferencias más notables se pueden observar en la Tabla 1.1 a continuación (Uñoja 2012):

Metodología tradicional	Metodología ágil
Está basada en normas provenientes de estándares guiados por el entorno de desarrollo	Está basada en heurísticas resultantes de la práctica de producción de código
Cierta resistencia a los cambios	Adaptable a los cambios
Impuestas externamente	Impuestas por el equipo
Proceso mucho más controlado	Proceso menos controlado
El cliente sólo interactúa con el equipo de desarrollo mediante reuniones	El cliente forma parte del equipo
Más artefactos	Pocos artefactos
Más roles	Pocos roles
Grupos grandes y distribuidos	Pequeños grupos de 2 a 10 personas y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe un contrato tradicional, o al menos es bastante flexible

Tabla 1.1. Comparación entre metodología tradicional y ágil

Una de las comparaciones que más influye en la selección de la metodología es que las ágiles deberían ser aplicadas en proyectos donde existe mucha incertidumbre y los requisitos no se conocen con exactitud, mientras que las tradicionales exigen al cliente tomar decisiones al inicio del proyecto.

## *Capítulo 1. Fundamentación Teórica*

Teniendo en cuenta las comparaciones antes planteadas, a pesar de la organización, control y planificación del proyecto que presentan las metodologías tradicionales; se seleccionó la metodología ágil haciendo énfasis en la adaptabilidad a los cambios y a la interacción entre el cliente y el equipo de desarrollo, lo cual permite obtener un producto más satisfactorio según las necesidades y especificaciones del cliente.

### **1.6.2.4. Metodología XP**

La programación extrema o eXtreme Programming (XP) es la más destacada de los procesos ágiles de desarrollo de software. Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

Tiene como características principales: desarrollo iterativo e incremental, programación en parejas, frecuente integración del equipo de programación con el cliente y simplicidad en el código (Letelier 2011).

### **1.6.2.5. Metodología Scrum**

La metodología Scrum es un método ágil de desarrollo muy simple que requiere de un duro trabajo al no estar basada en el seguimiento de un plan, sino a la continua adaptación a las circunstancias de la evolución del proyecto. Cumpliendo con las características de las metodologías ágiles se enfoca en construir el producto de forma incremental a través de breves iteraciones siguiendo la necesidad del cliente. Dichas iteraciones (sprints) se repiten continuamente hasta que el cliente da por concluido el producto. Scrum aplica un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto, está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto y los requisitos son cambiantes o poco definidos (López 2013).

Scrum reconoce que los clientes pueden cambiar de idea sobre lo que necesitan y los desafíos no pueden ser fácilmente enfrentados de manera planificada o predictiva, por lo cual acepta que no puede definir o entender completamente el problema y se centra

en maximizar la capacidad del equipo de hacer entregas rápidamente y responder a requisitos emergentes.

### 1.6.2.6. Metodología XP vs Scrum

Al hacer una comparación entre estas dos metodologías ágiles tan similares cabe resaltar que Scrum enfatiza más en el desarrollo de un proyecto a diferencia de XP que especifica más detalladamente las prácticas de la ingeniería, tales la programación en parejas o desarrollo basado en pruebas.

Otra de las diferencias es que Scrum no le permite al cliente hacer cambios durante la iteración hasta que esta no termine, mientras que XP suele ser más aceptable para introducir cambios durante la iteración.

Uno de los inconvenientes de Scrum es que supone que el cliente está muy involucrado en el desarrollo y revisa frecuentemente el avance de la funcionalidad, cuando en realidad el cliente participa pero no dedica tiempo y recursos para revisar pequeños avances en el desarrollo. Sin embargo en XP el cliente forma parte del equipo teniendo una fuerte comunicación con los desarrolladores.

Teniendo en cuenta las diferencias existentes entre estas metodologías se seleccionó XP como metodología de desarrollo de software.

## 1.7. Lenguajes de programación

A continuación se explicarán los lenguajes de programación que se proponen utilizar para el desarrollo de la Herramienta Informática.

### 1.7.1. Python

Python es un lenguaje de programación de alto nivel de propósito general cuyo diseño está enfocado a la legibilidad. Es un lenguaje interpretado, de tipos dinámicos y multiplataforma (Vigo 2012). Este lenguaje de programación soporta orientación a objetos, programación funcional, además de poseer una licencia de código abierto, denominada Python Software Foundation License compatible con la licencia pública general de GNU a partir de la versión 2.1.1.

Python resulta ser un lenguaje ágil y poderoso para representar casi cualquier tipo de solución teniendo en cuenta que en cuanto a sintaxis permite escribir código más

legible haciendo que se mantenga más a largo plazo. Además Python cuenta con librerías completas, robustas y bien documentadas.

## 1.7.2. PL/PgSQL

PL/pgSQL (Procedural Language/PostgreSQL Structured Query Language) es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL que permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas. Este lenguaje puede ejecutar programación en el servidor de base de datos teniendo como ventaja principal que las consultas y el resultado no tiene que ser transportadas entre el cliente y el servidor ya que los datos están guardados en el propio servidor.

Las funciones escritas en PL/pgSQL aceptan argumentos y pueden devolver valores tales como registros, vectores, conjuntos y tablas.

## 1.8. Entorno de Desarrollo Integrado

Eclipse fue desarrollado originalmente por IBM Canada y actualmente desarrollado por la *Fundación Eclipse* que es independiente y fomenta una comunidad de código abierto. La arquitectura de Eclipse basada en plugins soporta varios lenguajes, tales como: Java, Ruby, PHP, Python o Prolog.

Entre sus características principales se destaca por disponer de un editor de texto con resaltado de sintaxis donde puedes ver el contenido del fichero en el que estás trabajando, compilar en tiempo real, además de utilizar plugins, donde, a través de estos es posible añadir un sistema de control de versiones a través de Subversion y a la vez lograr una integración mediante Hibernate (EcuRed 2012).

Para desarrollar la Herramienta Informática que se propone se decidió utilizar Eclipse pues es una arquitectura destacada entre los entornos de desarrollo y se integra con el lenguaje de programación Python, además de funcionar en varias plataformas.

## 1.9. Interfaz gráfica de usuario

Para el desarrollo de la interfaz gráfica de la Herramienta Informática se propone utilizar la biblioteca PyQt dado que es desarrollado en el lenguaje de programación Python, el cual fue seleccionado anteriormente, y el programa QtDesigner para el diseño de la interfaz gráfica de usuario del framework Qt.

## 1.9.1. PyQt: Qt para Python

PyQt (Esquives 2011) es una adaptación de la biblioteca gráfica Qt para el lenguaje de programación Python, la cual está disponible para Windows, GNU/Linux y Mac OS X bajo diferentes licencias.

Qt es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario, constituyen actualmente una de las mejores en su tipo en el mundo del software libre. Realizadas en el lenguaje de programación C++, están hechas con el fin de realizar más con menos código (Esquives 2011).

Con la facilidad de Python sumada a la excelencia de Qt, se hace más agradable la programación visual. Cuenta con una amplia documentación proveniente de Qt y de Python a la vez (Leyva 2011).

Utilizando PyQt en el desarrollo de la Herramienta Informática se puede crear una interfaz visual sencilla y sin muchos contratiempos, ya que PyQt posee los componentes visuales necesarios para su desarrollo, así como una abundante documentación y ejemplos.

## 1.9.2. QtDesigner

QtDesigner es un programa para desarrollar interfaces gráficas de usuario del Framework Qt. Formas creadas con QtDesigner integra a la perfección con el código programado, con señales de Qt y el mecanismo de ranuras, que le permite asignar fácilmente el comportamiento de los elementos gráficos. Todas las propiedades establecidas en QtDesigner se puede cambiar de forma dinámica dentro del código (QtProject 2013).

QtDesigner se suele utilizar en conjunción con KDevelop, aunque es multiplataforma. Sirve para generar ficheros .ui, que contienen la interfaz gráfica de un programa que utilice el toolkit Qt. Gracias a las utilidades de la biblioteca Qt, se puede generar código automáticamente a partir de los ficheros creados con QtDesigner (Esquives 2011).

## 1.10. Conclusiones del capítulo

A partir del estudio realizado para dar solución a la problemática planteada, se arribó a las siguientes conclusiones:

- Se debe trabajar en un proyecto de Minería de datos aplicando un proceso de KDD.

## *Capítulo 1. Fundamentación Teórica*

- Se seleccionó la técnica descriptiva Reglas de asociación para la toma de decisiones teniendo en cuenta los resultados académicos de los estudiantes de la UCI en las asignaturas del departamento de Ingeniería de software impartidas hasta el quinto semestre y el rol asignado, con el objetivo de describir sus resultados.
- Se escogió como entorno de desarrollo Eclipse dada su integración al lenguaje de programación Python y por ser multiplataforma.
- Para el desarrollo de la Herramienta Informática se seleccionaron además el RapidMiner y Pentaho los cuales son de gran utilidad para aplicar el algoritmo de Minería de datos FP-Growth.
- Para guiar el proceso de descubrimiento de conocimiento en Bases de Datos se decidió seleccionar la Metodología CRISP-DM, así como la Metodología XP en el desarrollo de la Herramienta Informática propuesta.

# *Capítulo 2. Características de la herramienta informática propuesta*

## **2.1. Introducción**

En este capítulo se presentan las fases de la metodología CRISP-DM para enfrentar un proceso de KDD, con el objetivo de comprender la problemática antes expuesta. Se explican la Fase de exploración y Fase de planeamiento de la metodología XP necesarias para la comprensión y planificación de la Herramienta Informática y además se muestran las características requeridas del sistema.

## **2.2. Objeto de automatización**

La realización de esta Herramienta Informática tiene como objetivo automatizar un proceso KDD donde se asocien los resultados académicos de los estudiantes de la UCI en las asignaturas del departamento de ISW impartidas desde primero hasta el primer semestre de tercer año con el rol que desempeñan. Con esto se pretende apoyar la toma de decisiones en la asignación de roles de los estudiantes en los proyectos productivos. Se definen como procesos a automatizar:

- Preparar Datos.
- Aplicar Minería.
- Difundir Resultados.

Se inicia con la preparación de los datos necesarios, posteriormente se aplica la Minería de datos obteniendo los resultados de este proceso, donde el usuario tiene la opción de difundir los resultados. En caso positivo difunde los resultados y puede probar con otros datos iniciando nuevamente el proceso de preparación de los datos o salir de la aplicación. Si no desea difundir los resultados se puede intentar con otros datos y regresar al primer proceso o salir de la aplicación. En la Figura 2.1 se muestra cómo se despliegan estos procesos:

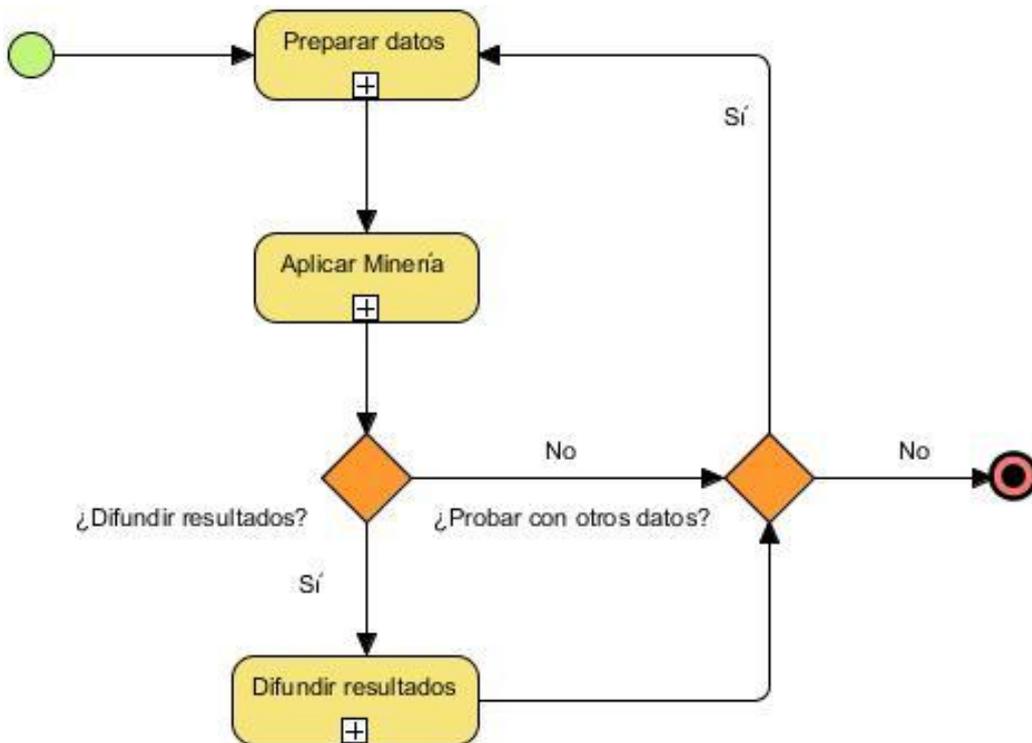


Figura 2.1. Diagrama del proceso general del negocio.

### 2.3. Solución propuesta

Se decidió construir una Herramienta Informática que solucione la problemática antes expuesta de forma automática, la cual es una herramienta de escritorio que integra programas y tecnologías multiplataforma. Dicha herramienta permite llevar a cabo un proceso KDD donde: selecciona, integra, limpia y transforma los datos, además de aplicar la tarea descriptiva de Minería de datos Reglas de asociación, la visualización y difusión de los resultados obtenidos. Esta Herramienta Informática se decidió nombrarla HADOC.

### 2.4. Fases de CRISP-DM

A continuación se muestra el despliegue de cada fase de la metodología CRISP-DM como guía del proceso KDD para el desarrollo de esta Herramienta Informática.

### 2.4.1. Comprensión del negocio

Para la comprensión del negocio es necesario definir el problema como un proyecto de Minería de datos que explique los procesos para realizar la Herramienta Informática propuesta.

La implementación de la herramienta tiene como objetivo fundamental obtener reglas de asociación a partir de las notas de los estudiantes en asignaturas impartidas por el departamento de ISW de la Facultad 2. Estas asignaturas se vinculan con el rol que desempeña el estudiante en un proyecto productivo. Se pretende encontrar patrones que describan cómo se está comportando la asignación de roles en los proyectos productivos a nivel de centro. La Facultad cuenta con el Centro Telemática (TLM) así como con el Centro de Informatización para la Seguridad Ciudadana (ISEC). La plataforma Gestión de Proyectos provee toda la información que vincula a cada estudiante con el rol asignado en cada proyecto productivo, de la cual se obtuvo una muestra con estos datos. Las notas de todos los estudiantes vinculadas con sus respectivas asignaturas se encuentran almacenadas en la Plataforma Gestión Universitaria de la cual se obtuvo una muestra de 568 estudiantes de la Facultad 2.

Toda la información necesaria de las fuentes de datos se prepara y se almacena luego en la base de datos de la Herramienta para la etapa de minería con la ayuda del software Pentaho.

Los softwares seleccionados para apoyar el desarrollo de las funcionalidades de la Herramienta HADOC han sido utilizados a través de líneas de comandos, eliminando así la interacción directa del usuario con estos. Las transformaciones a realizar en Pentaho así como los procesos en RapidMiner, son creados por HADOC gracias a la tecnología *XML scripting* que poseen ambos softwares.

### 2.4.2. Comprensión de los datos

La muestra obtenida proveniente de las fuentes de datos contenía información irrelevante, incluso existían estudiantes que no tenían actualizado en la muestra el rol asignado en un proyecto y otros con más de un rol en otros casos. Algunos de los estudiantes no contenían las notas de todas las asignaturas cursadas por los mismos. En total se obtuvieron las notas de 78 asignaturas vinculadas con el nombre y

apellidos 568 estudiantes. Estas estaban expresadas con valores enteros de 3,4 y 5 siendo este último la nota máxima alcanzada, quedando campos vacíos donde se desconocía la nota oficial.

### 2.4.3. Preparación de los Datos

Para la selección de los atributos a minar se utilizaron las notas de 8 asignaturas pertenecientes al departamento de ISW de la Facultad. Se obtuvieron además el nombre de los estudiantes, campo que no aparece en la vista minable, pero necesario para actualizar en la base de datos de HADOC el rol asignado a cada estudiante en la Plataforma Gestión de Proyectos. Las asignaturas seleccionadas son las cursadas en el transcurso de la carrera hasta el quinto semestre, fecha en la cual se comienza la vinculación de los estudiantes a proyectos productivos. Dada la necesidad de utilizar atributos binominales para el algoritmo FP-Growth seleccionado para la búsqueda de ítems frecuentes en la etapa de minería, se realiza un proceso de preparación de datos antes del almacenado de los datos en la Herramienta con la ayuda de Pentaho. De este software se utilizan los siguientes operadores:

- **Table Input:** operador para la extracción de datos a través de una conexión previamente configurada. El proceso de filtrado de las asignaturas antes seleccionadas en la vista de preparación de datos se realiza construyendo una consulta dinámicamente. Este operador provee un campo para ejecutar consultas a la base de datos de la conexión configurada.
- **Value Mapper:** este operador se utilizó para transformar atributos enteros a nominales.
- **Table Output:** se utilizó para guardar toda la información ya transformada en la base de datos de HADOC.

Existen varios métodos para el tratamiento de campos vacíos, (ejemplo: la media de los valores existentes, valores según el negocio, etc.) todos estos responsables de introducir valores irreales (ruido) a la información obteniendo resultados ficticios en la generación de reglas. Se decidió no darle un tratamiento a los campos vacíos, teniendo como desventaja posibles reglas de asociación perdidas pero en cambio se obtienen reglas verídicas dadas la información existente.

### 2.4.4. Modelado

Una vez terminada la fase de preparación con la herramienta Pentaho es necesario aún utilizar un último operador, esta vez de la herramienta RapidMiner. Para la obtención de reglas de asociación se utilizaron los siguientes operadores:

- **Table Input:** necesario para la extracción de la información almacenada en la base de datos de HADOC para su posterior procesado.
- **Nominal to Binominal:** este operador transforma los atributos nominales a binominales, operación necesaria para la entrada de datos del algoritmo seleccionado.
- **FP-Growth:** algoritmo seleccionado para la obtención de ítems frecuentes.
- **Create Rules:** tiene como entrada los ítems frecuentes generados por FP-Growth para crear reglas de asociación.
- **Write as Text:** exporta en formato *txt* un archivo con las reglas de asociación generadas a partir del proceso llevado a cabo.

En busca de las reglas con mayor confianza, la etapa de minería se comienza con un soporte mínimo = 0.1%, para generar la mayor cantidad de ítems frecuentes posibles. Para la generación de reglas, se comienza con un valor mínimo de 0.05%. Esta variable aumenta hasta que existan 100 reglas de asociación con la mayor confianza generadas por la Herramienta Informática, nunca sobrepasando este umbral y a veces devolviendo menos reglas que las deseadas en correspondencia con la cantidad de atributos seleccionados en la vista minable.

### 2.4.5. Evaluación

En esta fase se visualizan e interpretan los patrones generados por la Herramienta Informática, removiendo de ser necesarios, patrones redundantes o irrelevantes, así como la traducción de los patrones útiles en términos que sean entendibles para el usuario.

Ejemplo de una regla interesante generado por la herramienta:

[pp 1 = Excelente, pp 2 = Excelente] --> [rol = Programador] (confidence: 0.763)

Esta dispone que el estudiante que ha obtenido Excelente en (Práctica Profesional 1 y 2, pp 1 y pp 2 respectivamente) está realizando el rol de Programador en un proyecto productivo.

### 2.4.6. Despliegue

Para la fase de despliegue se genera de forma opcional un archivo en formato *pdf* con el nombre "Informe.pdf" que contiene todas las reglas de asociación luego de la fase de evaluación y sólo se almacenan en una tabla dinámicamente.

También se puede enviar de opcionalmente el archivo a los interesados vía correo electrónico, como se puede observar en el [Anexo 1](#).

## 2.5. Requerimientos no funcionales

Los requerimientos no funcionales constituyen las características requeridas del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señale una restricción del mismo. Describe no lo que el software hará, sino cómo lo hará, suelen ser más críticos que los funcionales, difíciles de verificar además deben ser apropiadamente definidos, analizados y trazados (Córcoles 2011).

#### ➤ Apariencia o interfaz externa

- La Herramienta Informática debe presentar una interfaz agradable, sencilla y fácil de usar, de forma tal que el usuario pueda explotar al máximo las funcionalidades que brinda para agilizar su trabajo.
- Las interfaces deben contar con los componentes visuales necesarios para las operaciones correspondientes, evitando la sobrecarga de imágenes, tales como: autenticar usuario, aplicar Minería de datos, etc.
- Las interfaces deben contar con letras legibles y colores agradables a la vista, sin información repetida, ni en exceso.

#### ➤ Usabilidad

- El sistema podrá ser usado por personas que tienen o no, habilidades en el trabajo con la computadora, debido a esto está estructurado de forma sencilla.

## Capítulo 2. Características de la herramienta informática

### *propuesta*

- Es necesario garantizar un fácil acceso de los usuarios a la Herramienta Informática, donde solo puedan usar la herramienta el personal autorizado.

#### ➤ **Soporte**

- Cuando sea desplegada la Herramienta Informática se realizará un mantenimiento total del producto en cada curso que incluirá la actualización de los datos seleccionados.

#### ➤ **Requerimientos de hardware**

- Como mínimo de prestaciones CPU de 1GHz y 512mb de RAM.

#### ➤ **Requerimientos de software**

- Es necesaria la previa instalación del intérprete para el lenguaje Python así como de la versión 1.6.0\_24 de la máquina virtual de java.
- Se debe tener instalado y configurado un servidor de base de datos PostgreSQL.

#### ➤ **Restricciones en el diseño y la implementación**

- La versión del lenguaje Python 2.6.
- La codificación será la determinada en la guía de estilo para el lenguaje Python.
- Para la codificación se utilizará Eclipse, facilitando considerablemente la programación con Python en el mismo.
- Han de estar instaladas las bibliotecas de Python:
  - ❖ python-dev
  - ❖ python-qt4
  - ❖ pyqt4-dev-tools
  - ❖ python-qt4-dev
  - ❖ python-psycog2
  - ❖ python-SOAPpy
  - ❖ python-reportlab

#### ➤ **Requerimientos de seguridad**

- **Disponibilidad:** el sistema debe de estar disponible siempre que se necesite ejecutarlo. En caso de no encontrarse disponible alguna de las fuentes de datos la Herramienta Informática debe informar al usuario esta situación.
- **Confidencialidad:** el usuario sólo puede entrar a la Herramienta Informática autenticándose.
- **Integridad:** la contraseña de todas las conexiones a las bases de datos están encriptadas mediante un algoritmo de encriptación de 64 bites. Este permite encriptar las contraseñas para almacenarlas y desencriptarlas cuando se realicen las funciones de transformación de los datos y de aplicar minería.

### 2.6. Usuarios de la Herramienta Informática

Los usuarios del sistema son quienes administran las funcionalidades y obtienen resultados, producto de algún proceso dentro de la Herramienta Informática.

Administrador: usuario encargado de realizar la preparación y transformación de los datos.

Usuario: se define como los usuarios que necesitan tomar decisiones a partir de los datos de las asignaturas del departamento de ISW impartidas por los estudiantes hasta el quinto semestre así como los roles que estos tienen asignados, en este caso los usuarios pudieran ser: jefe del centro, líder de proyecto y la vicedecana docente.

Si el usuario que accede a la Herramienta Informática tiene rol Administrador, puede acceder a la vista Administración HADOC con las funcionalidades de almacenar y cargar las configuraciones de las conexiones a las Bases de datos, preparación de los datos y gestionar usuario.

En caso de tener un rol usuario, accede a la vista de HADOC con las funcionalidades de aplicar minería, visualizar resultados y difundir los resultados.

### 2.7. Fase de exploración

En esta primera fase de la metodología XP son planteadas por los clientes las funciones que debe realizar la Herramienta Informática, así como las Historias de Usuario de mayor interés para una primera entrega del producto, las cuales describen

las características y funcionalidades requeridas para el software que se va a elaborar (Ramírez 2008). En esta etapa el equipo de desarrollo va conociendo y familiarizándose con las herramientas que se van a utilizar en el desarrollo del proyecto.

### 2.7.1. Lista de funciones

1. **Autenticar usuario:** identifica a los usuarios y los autoriza a utilizar las funcionalidades de la Herramienta Informática según el rol que desempeñen.
2. **Gestionar usuario:** el administrador gestiona los usuarios que tengan acceso a la Herramienta Informática.
  - 2.1. **Adicionar usuario:** permite adicionar usuarios para que tengan acceso a la Herramienta Informática.
  - 2.2. **Modificar usuario:** permite modificar los datos de los usuarios que acceden a la Herramienta Informática.
  - 2.3. **Eliminar usuario:** elimina usuarios con acceso a la Herramienta Informática.
3. **Almacenar configuración de conexiones:** almacena las configuraciones de las conexiones a las bases de datos y al servidor de correo.
4. **Cargar configuración de conexiones:** obtiene las configuraciones de las conexiones previamente almacenadas.
5. **Preparar datos:** se obtienen y se filtran de la base de datos Gestión Universitaria las asignaturas del departamento de ISW impartidas a los estudiantes hasta el quinto semestre, así como el rol asignado de estos estudiantes en GESPRO. Dichos datos son almacenados en la base de datos de HADOC.
6. **Aplicar minería:** obtiene Reglas de asociación de las opciones: asignaturas o asignaturas y roles, seleccionadas por el usuario.
7. **Difundir resultados:** permite enviar el reporte de los resultados de forma opcional en un mensaje de correo electrónico a los interesados.
8. **Generar reporte en pdf:** genera reporte en formato *pdf* de los resultados obtenidos al aplicar minería.

### 2.7.2. Historias de Usuario

Las Historias de Usuario (HU) representan una breve descripción del comportamiento del sistema, son una forma breve de administrar los requisitos de los usuarios sin elaborar grandes cantidades de documentos. Se realiza una por cada característica principal del sistema (Freddy 2012).

Dentro de la metodología XP estas son escritas por los clientes, el cual es el encargado de darle una prioridad a cada historia de usuario y el equipo de desarrollo es quien evalúa las historias y les da un costo, este se traduce en las semanas que se llevará el desarrollo de la misma.

Si la historia requiere más de tres semanas de desarrollo se les sugiere al cliente que las divida en pequeñas historias de usuario, dándoles a estas nuevas una prioridad y un costo. También es importante destacar que las historias de usuarios nuevas pueden describirse en cualquier momento, con esto se comprueba la flexibilidad de la metodología.

A continuación se muestran las Historias de Usuario necesarias para la realización de la Herramienta Informática.

Historia de Usuario	
<b>Número:</b> 1	<b>Nombre de Historia de Usuario:</b> Autenticar usuario
<b>Modificación de HU Número:</b>	
<b>Usuario:</b> Administrador, Usuario	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> alta	<b>Riesgo en Desarrollo:</b> alto
<b>Puntos Estimados:</b> 1	<b>Puntos Reales:</b> 1
<b>Descripción:</b> identifica a los usuarios y los autoriza a utilizar las funcionalidades de la Herramienta Informática según el rol que desempeñen.	
<b>Observaciones:</b>	



Tabla 2.1. HU. Autenticar usuario

Historia de Usuario	
Número: 2	Nombre de Historia de Usuario: Adicionar usuario
Modificación de HU Número:	
Usuario: Administrador	Iteración Asignada: 1
Prioridad en Negocio: media	Riesgo en Desarrollo: bajo
Puntos Estimados: 2	Puntos Reales: 2
Descripción: permite adicionar usuarios para que tengan acceso a la Herramienta Informática.	
Observaciones:	
<p><b>Prototipo de interfaz</b></p> <p>El prototipo de interfaz que se muestra a continuación representa el Gestionar usuario donde se realizan las funcionalidades de Adicionar, Modificar y Eliminar usuario.</p>	

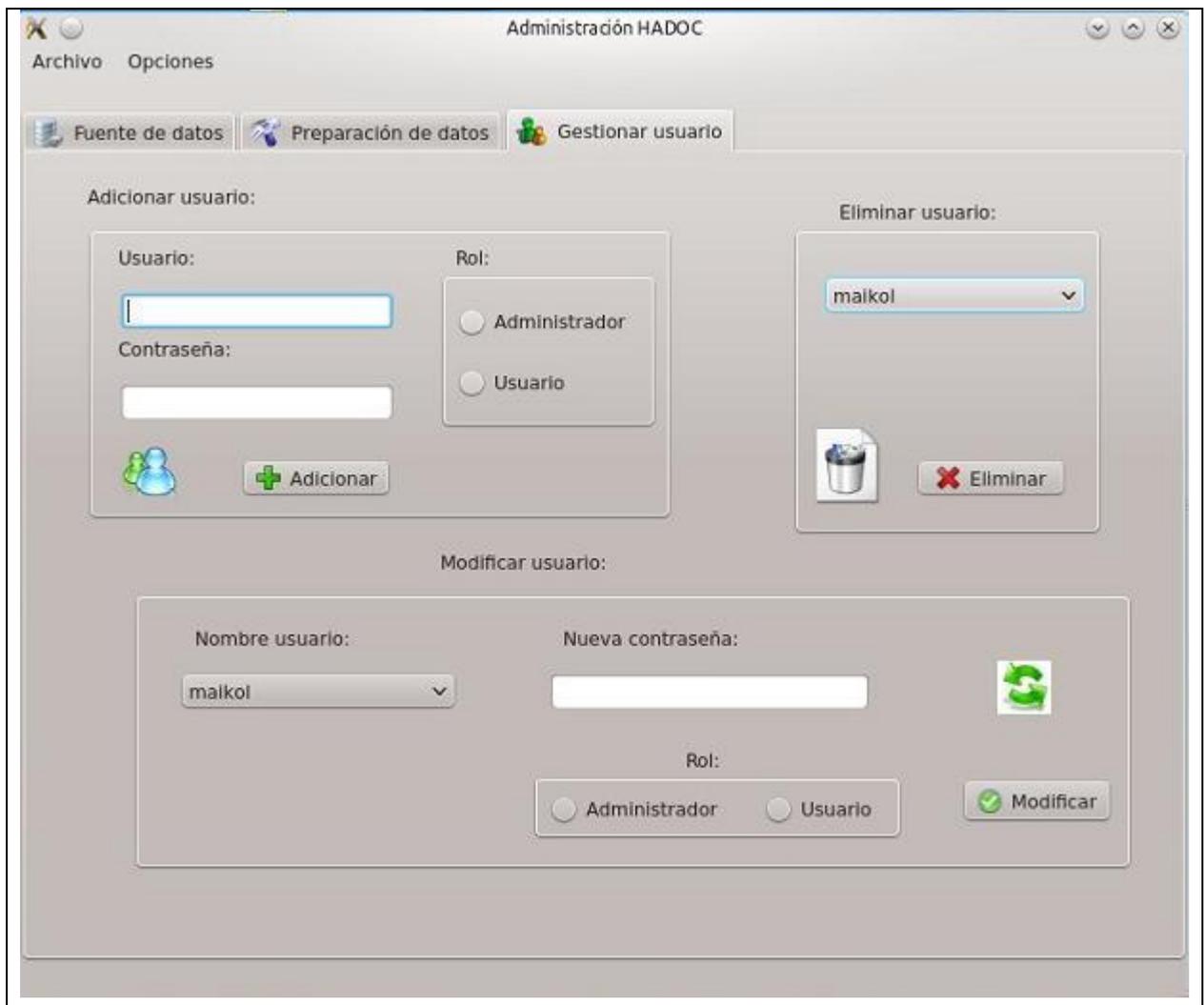


Tabla 2.2. HU. Adicionar usuario

Historia de Usuario	
Número: 3	Nombre de Historia de Usuario: Modificar usuario
Modificación de HU Número:	
Usuario: Administrador	Iteración Asignada: 1
Prioridad en Negocio: media	Riesgo en Desarrollo: bajo
Puntos Estimados: 2	Puntos Reales: 2
Descripción: permite modificar los datos de los usuarios que acceden a la Herramienta Informática.	
Observaciones:	

## Capítulo 2. Características de la herramienta informática

*propuesta*

**Prototipo de interfaz.** Ver HU Adicionar usuario.

Tabla 2.3. HU. Modificar usuario

Historia de Usuario	
<b>Número:</b> 4	<b>Nombre de Historia de Usuario:</b> Eliminar usuario
<b>Modificación de HU Número:</b>	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> media	<b>Riesgo en Desarrollo:</b> bajo
<b>Puntos Estimados:</b> 2	<b>Puntos Reales:</b> 2
<b>Descripción:</b> elimina usuarios con acceso a la Herramienta Informática.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz.</b> Ver HU Adicionar usuario.	

Tabla 2.4. HU. Eliminar usuario

Historia de Usuario	
<b>Número:</b> 5	<b>Nombre de Historia de Usuario:</b> Almacenar configuración de conexiones
<b>Modificación de HU Número:</b>	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> media	<b>Riesgo en Desarrollo:</b> alto
<b>Puntos Estimados:</b> 3	<b>Puntos Reales:</b> 3
<b>Descripción:</b> almacena las configuraciones de las conexiones a las bases de datos y al servidor de correo.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz</b>	
<p>En la siguiente interfaz se muestra la pestaña Opciones donde se brindan las funcionalidades de Almacenar y Cargar las configuraciones de las conexiones al Sistema de Gestión Universitaria, al Servidor GESPRO, a la BD de la Herramienta Informática propuesta y al Servidor de Correo.</p>	

## Capítulo 2. Características de la herramienta informática

*propuesta*

The screenshot shows the 'Administración HADOC' application window with a menu bar containing 'Archivo' and 'Opciones'. Below the menu bar are three tabs: 'Fuente de datos', 'Preparación de datos', and 'Gestionar usuario'. The main area is divided into four sections:

- Conexión Gestión Universitaria:** Includes input fields for 'Servidor:', 'Puerto:', 'Nombre BD:', 'Usuario:', and 'Contraseña:', along with a 'Conexión' button and a refresh icon.
- Conexión BD HADOC:** Includes input fields for 'Servidor:', 'Puerto:', 'Nombre BD:', 'Usuario:', and 'Contraseña:', along with a 'Conexión' button and a refresh icon.
- Servidor GESPRO:** Includes input fields for 'Usuario:', 'Servidor:', and 'Contraseña:', along with a 'Conexión' button and a refresh icon.
- Credenciales Servidor Correo:** Includes input fields for 'Remitente:' and 'Contraseña:', along with a 'Conexion' button and a refresh icon.

Tabla 2.5. HU. Almacenar configuración de conexiones

Historia de Usuario	
Número: 6	Nombre de Historia de Usuario: Cargar configuración de conexiones
Modificación de HU Número:	
Usuario: Administrador	Iteración Asignada: 1
Prioridad en Negocio: media	Riesgo en Desarrollo: alto
Puntos Estimados: 3	Puntos Reales: 3
Descripción: obtiene las configuraciones de las conexiones previamente almacenadas.	
Observaciones:	
Prototipo de interfaz:	

## Capítulo 2. Características de la herramienta informática

*propuesta*

The screenshot shows the 'Administración HADOC' application window. At the top, there is a menu bar with 'Archivo' and 'Opciones'. Below the menu bar are three tabs: 'Fuente de datos', 'Preparación de datos', and 'Gestionar usuario'. The main content area is divided into four sections:

- Conexión Gestión Universitaria:** Includes fields for 'Servidor:' (localhost), 'Puerto:' (5432), 'Nombre BD:' (gestUniv), 'Usuario:' (postgres), and 'Contraseña:' (masked). A 'Conexión' button and a green checkmark icon are at the bottom.
- Conexión BD HADOC:** Includes fields for 'Servidor:' (localhost), 'Puerto:' (5432), 'Nombre BD:' (HADOC), 'Usuario:' (postgres), and 'Contraseña:' (masked). A 'Conexión' button and a green checkmark icon are at the bottom.
- Servidor GESPRO:** Includes fields for 'Usuario:' (ybelen), 'Servidor:' (localhost), and 'Contraseña:' (masked). A 'Conexión' button and a green checkmark icon are at the bottom.
- Credenciales Servidor Correo:** Includes fields for 'Remitente:' (ybelen@estudiantes.uci.cu), 'Conexión' button, and 'Contraseña:' (masked). A green checkmark icon is at the bottom.

Tabla 2.6. HU. Cargar configuración de conexiones

Historia de Usuario	
<b>Número:</b> 7	<b>Nombre de Historia de Usuario:</b> Preparar datos
<b>Modificación de HU Número:</b>	
<b>Usuario:</b> Administrador	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> alta	<b>Riesgo en Desarrollo:</b> alto
<b>Puntos Estimados:</b> 3	<b>Puntos Reales:</b> 3
<b>Descripción:</b> se obtienen y se filtran de la base de datos Gestión Universitaria las asignaturas del departamento de ISW impartidas a los estudiantes hasta el quinto semestre, así como el rol asignado de estos estudiantes en GESPRO. Dichos datos son almacenados	

en la base de datos de HADOC.

**Observaciones:**

**Prototipo de interfaz:**



Tabla 2.7. HU. Preparar datos

Historia de Usuario	
Número: 8	Nombre de Historia de Usuario: Aplicar minería
Modificación de HU Número:	
Usuario: Usuario	Iteración Asignada: 2
Prioridad en Negocio: alta	Riesgo en Desarrollo: alto
Puntos Estimados:3	Puntos Reales: 3

**Descripción:** obtiene Reglas de asociación de las opciones: asignaturas o asignaturas y roles, seleccionadas por el usuario.

**Observaciones:**

**Prototipo de interfaz**



Tabla 2.8. HU. Aplicar minería

Historia de Usuario	
Número: 9	Nombre de Historia de Usuario: Difundir resultados
Modificación de HU Número:	
Usuario: Usuario	Iteración Asignada: 3
Prioridad en Negocio: alta	Riesgo en Desarrollo: alto

## Capítulo 2. Características de la herramienta informática

*propuesta*

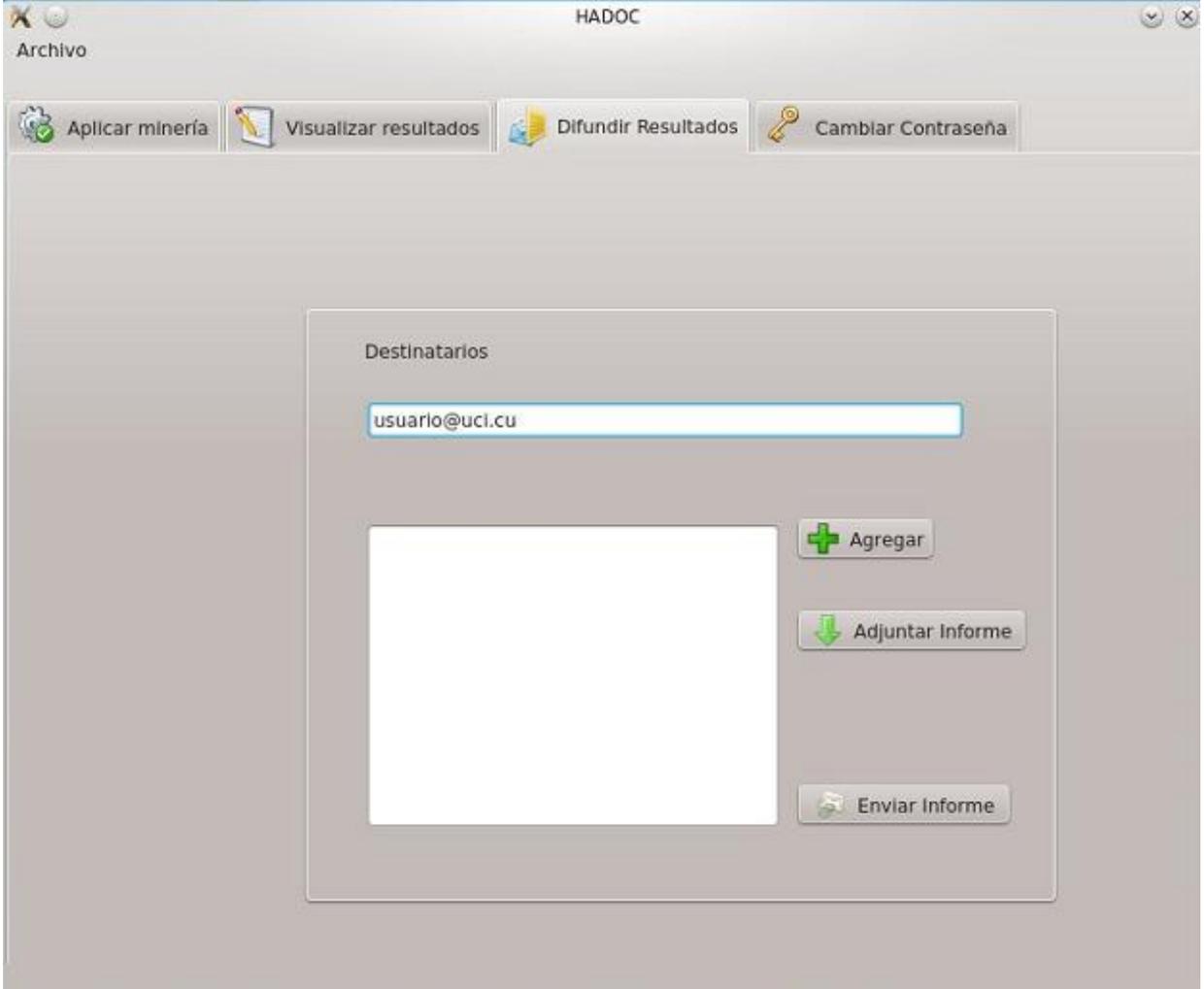
<b>Puntos Estimados:</b> 3	<b>Puntos Reales:</b> 3
<b>Descripción:</b> permite enviar el reporte de los resultados de forma opcional en un mensaje de correo electrónico a los interesados.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz</b>	
	

Tabla 2.9. HU. Difundir resultados

Historia de Usuario	
<b>Número:</b> 10	<b>Nombre de Historia de Usuario:</b> Generar reporte en <i>pdf</i> .
<b>Modificación de HU Número:</b>	
<b>Usuario:</b> Usuario	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> media	<b>Riesgo en Desarrollo:</b> bajo

<b>Puntos Estimados:</b> 2	<b>Puntos Reales:</b> 2
<b>Descripción:</b> genera reporte en formato <i>pdf</i> de los resultados obtenidos al aplicar minería.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz</b>	
 <p>The screenshot shows a software window titled 'HADOC' with a menu bar containing 'Archivo'. Below the menu bar is a toolbar with four icons: 'Aplicar minería', 'Visualizar resultados', 'Difundir Resultados', and 'Cambiar Contraseña'. The main area displays a 'Reporte de Resultados' window with a list of results. Each result line includes a confidence score in parentheses. At the bottom of the report window is a 'Guardar como PDF' button.</p>	

Tabla 2.10. HU. Generar reportes en *pdf*

## 2.8. Fase de planeamiento

En la fase de planeamiento se priorizan las historias de usuario y los programadores estiman cuánto esfuerzo requiere cada historia para definir el cronograma. Para medir el esfuerzo se utiliza como medida el punto, el cual expresa una semana de trabajo sin interrupciones. En esta fase se aplicarán además diferentes iteraciones determinadas tanto por el cliente como por los desarrolladores, donde con cada iteración se debe entregar una nueva versión al cliente. Cada iteración es planificada con las historias de

usuario esenciales. Es importante hacer muchas entregas y frecuentes lo que permite detectar posibles errores. Se estima también la velocidad del equipo por los desarrolladores y el plan de entrega es elaborado por los clientes y desarrolladores (Ramírez 2008).

### 2.8.1. Estimación de esfuerzos

Es necesario realizar una estimación del tiempo de trabajo para la realización de la Herramienta Informática, para esto se debe considerar el tiempo de cada una de las historias de usuario identificadas anteriormente que se muestran en la tabla a continuación:

Historia de usuario	Puntos de estimación
Autenticar usuario	1
Adicionar usuario	2
Modificar usuario	2
Eliminar usuario	2
Almacenar configuración de conexiones	3
Cargar configuración de conexiones	3
Preparar datos	3
Aplicar minería	3
Difundir resultados	3
Generar reportes en pdf	2

Tabla 2.11. Estimación de esfuerzos de cada historia de usuario

Los puntos de estimación proporcionados a cada HU expresan la cantidad de semanas estimadas para la implementación de estas. Teniendo el tiempo de duración de todas las HU se puede apreciar el tiempo total para la implementación de la Herramienta Informática.

## 2.9. Fase de iteraciones

Una vez determinadas las historias de usuario y estimado el esfuerzo de cada una de estas, se definen las iteraciones necesarias para la obtención de pequeñas versiones iterativas. Para cada iteración es necesario estimar el tiempo de duración y establecer

el orden en que se implementarán las historias de usuario, como se muestra en la Tabla 2.12.

Iteraciones	Orden de la HU	Duración de las iteraciones
Iteración 1	Autenticar usuario	16 semanas
	Adicionar usuario	
	Modificar usuario	
	Eliminar usuario	
	Almacenar configuración de conexiones	
	Cargar configuración de conexiones	
	Preparar datos	
Iteración 2	Aplicar minería	3 semanas
Iteración 3	Difundir resultados	5 semanas
	Generar reportes en <i>pdf</i>	
Tiempo Total	24 semanas	

Tabla 2.12. Plan y duración de las iteraciones

### 2.10. Conclusiones del capítulo

- Se desplegaron las fases de la metodología CRISP-DM para guiar el proceso KDD, describiendo en detalles todo el proceso para la realización de la Herramienta Informática.
- Se definieron el listado de funciones y los requisitos no funcionales con el objetivo de desarrollar la Herramienta Informática cumpliendo las especificaciones del cliente.
- Se desarrollaron las fases de exploración y de planificación para llevar a cabo las definiciones de las HU generando un total de diez HU descompuestas en tres iteraciones a implementar obteniendo un total de estimación de esfuerzo de 24 semanas.

# *Capítulo 3. Diseño de la herramienta informática propuesta*

## **3.1. Introducción**

En el presente capítulo se presenta el diseño para el desarrollo de la Herramienta Informática propuesta, teniendo en cuenta los patrones de diseño y de asignación de responsabilidades, la estructura del proyecto, el diseño de la BD así como la arquitectura propuesta para la organización lógica y física de la herramienta.

## **3.2. Propuesta de arquitectura de la Herramienta Informática**

La arquitectura del software de un programa o sistema de cómputo es la estructura o las estructuras del sistema, que incluyen los componentes del software, las propiedades visibles externamente de esos componentes y las relaciones entre ellos (Pressman 2007).

Para la arquitectura de la Herramienta Informática propuesta se decidió utilizar el patrón arquitectónico Arquitecturas en capas que se clasifica en diferentes estilos arquitectónicos dentro de los cuales se escogió la Arquitectura en tres capas para la organización del diseño de la herramienta.

Dicha arquitectura está organizada jerárquicamente en capas donde cada una provee servicios a la capa superior y sirve a la capa inferior, de esta forma la interacción está limitada a las capas adyacentes.

Este estilo arquitectónico presenta la ventaja de que se lleva a cabo el desarrollo en diferentes capas permitiendo que, en caso de ocurrir algún error o de realizarse un cambio, solo sea necesario cambiar la capa en cuestión sin afectar el correcto funcionamiento del resto de la Herramienta Informática.

Esta arquitectura está dividida en las siguientes capas:

## *Capítulo 3. Diseño de la herramienta informática propuesta*

**Capa de presentación:** consiste en la interfaz gráfica de usuario y es la encargada de la interacción de este con la herramienta y viceversa. Esta capa proporciona la información de las funcionalidades de HADOC al usuario.

**Capa de negocio:** en esta capa se encuentran las funcionalidades de la Herramienta Informática tales como: preparación de los datos, realizar transformación, aplicar minería, generar reportes, etc. Se reciben las peticiones del usuario, se procesa la información y se envían las respuestas a la capa de presentación mostrándolas en la interfaz de HADOC. Esta es la capa intermedia que tiene el código al que accede la capa de presentación para obtener los datos deseados de las bases de datos de HADOC, Gestión Universitaria, GESPRO, así como al servidor de correo.

**Capa de acceso a datos:** Es la encargada de gestionar el almacenamiento de los datos de la Herramienta Informática y de los usuarios, así como de devolver los datos a la capa intermedia. Esta interactúa con las bases de datos de HADOC, Gestión Universitaria y GESPRO.

Aunque esta arquitectura pueda referirse tanto en tres capas como en tres niveles, estos términos no significan lo mismo; las capas son la forma de representar la vista lógica, mientras que los niveles son las formas en que las capas están distribuidas físicamente. Por lo tanto la arquitectura de esta Herramienta Informática está conformada por tres capas y un nivel ya que todas estas capas se encuentran en una misma computadora.

### **3.3. Patrones**

Los patrones son una solución ya probada y documentada a un problema en un determinado contexto o situación. Se puede asegurar que un sistema para que quede bien estructurado debe estar lleno de patrones. Cada patrón hace una descripción de la solución de cada problema, y de esa forma la solución se puede utilizar una y otra vez, sin repetir las mismas acciones (Ricardo 2011).

#### **3.3.1. Patrones de diseño GoF (Gang of Four)**

A continuación se muestran los patrones de diseño identificados para el desarrollo de la Herramienta Informática:

**Diseño arquitectónico Modelo-Vista-Controlador de PyQT4:** PyQt4 contiene el módulo de Python QtGui. QtGui contiene la mayoría de las clases de interfaz gráfica de usuario.

## Capítulo 3. Diseño de la herramienta informática propuesta

### Patrones de creación:

- **Abstract Factory:** proporciona una interfaz para crear familias de objetos o que dependen entre sí, sin especificar sus clases concretas.

Ejemplo Interfaces gráficas: un caso relativamente común de uso de este patrón se da en la creación de familias de interfaces gráficas en las cuales los elementos (productos) del interfaz se mantienen constantes (por ejemplo labels, botones, cajas de texto...) pero el dibujado de dichos elementos puede delegarse en distintas familias (por ejemplo QT, GTK, etc.) de forma que, en función de la fábrica seleccionada obtenemos unos botones u otros.

### Patrones de Estructurales:

- **Bridge:** un puente tiene dos componentes esenciales, la "abstracción", que es el objeto responsable de inicializar una operación, y la "implementación", que es el objeto que la lleva a cabo. La abstracción conoce su implementación porque guarda una referencia a la misma, o porque lo posee. Frecuentemente ocurre que la abstracción crea la implementación, que no es un requerimiento absoluto del patrón.

Ejemplo de utilización: el botón conexión de la Vista Administración inicializa la operación de conectar con una base de datos enviando un mensaje cuando finaliza.

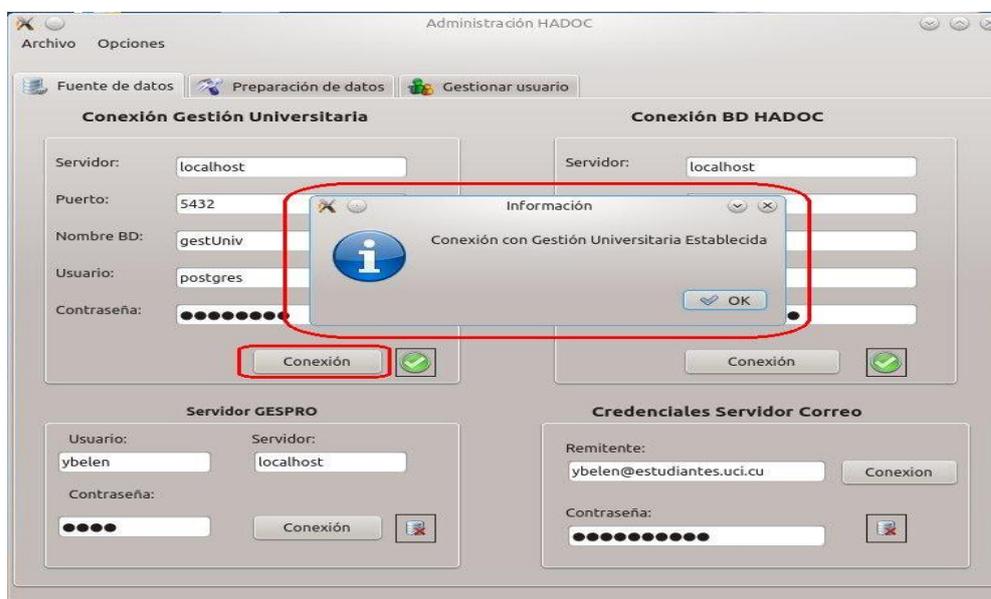


Figura 3.1. Ejemplo de utilización de Bridge

## Capítulo 3. Diseño de la herramienta informática propuesta

- **Facade (Fachada):** el patrón fachada viene motivado por la necesidad de estructurar un entorno de programación y reducir su complejidad con la división en subsistemas, minimizando las comunicaciones y dependencias entre éstos.

Ejemplo: a veces las bibliotecas son difícilmente legibles. Este patrón estructural resuelve la necesidad de acceder a un conjunto de APIs, creando un intermediario a un nivel menos abstracto para el desarrollador.

### Patrones de comportamiento:

- **Memento:** representa y externaliza el estado interno de un objeto sin violar la encapsulación, de forma que éste puede volver a dicho estado más tarde. Se usa este patrón cuando se quiere poder restaurar el sistema desde estados pasados y por otra parte, es usado cuando se desea facilitar el hacer y deshacer de determinadas operaciones, para lo que habrá que guardar los estados anteriores de los objetos sobre los que se opere (o bien recordar los cambios de forma incremental).

Ejemplo: el IDE Eclipse permite volver a estados anteriores del sistema.

- **Interpreter:** dado un lenguaje, define una representación de su gramática junto con un intérprete que usa dicha representación para interpretar las sentencias del lenguaje.

Ejemplo: lenguaje Python y su intérprete.

### 3.3.2. Patrones para Asignar Responsabilidades (GRASP)

A continuación se muestran los patrones GRASP identificados para el desarrollo de la Herramienta Informática:

**Experto:** su uso permite que las clases controladoras del sistema manejen las peticiones del cliente, las clases del negocio realicen las validaciones correspondientes de dichas peticiones y las de acceso a datos se encarguen de la gestión de los datos entregados desde la lógica del negocio, esto proporciona que se conserve el encapsulamiento y se soporte el bajo acoplamiento y la alta cohesión.

Ejemplo del empleo de este patrón en la Herramienta Informática se puede observar en la clase *Controladora\_Vista\_Administrador* quien maneja las peticiones del usuario para la preparación de los datos, la autenticación y la gestión de usuarios.

## Capítulo 3. Diseño de la herramienta informática propuesta

**Creador:** es el responsable de la creación de una nueva instancia de alguna clase.

Cada clase del negocio debe contener una instancia de su acceso a datos correspondiente y cada clase controladora debe tener una instancia de su negocio, garantizando que se cumpla con la correcta utilización de la arquitectura en capas.

Este patrón se puede evidenciar en la clase *CrearTransformacion* que crea una instancia de la clase *CrearXML* la cual tiene todos los pasos a seguir para realizar la transformación de los datos.

**Alta cohesión:** es el responsable de mantener la complejidad manejable. Este patrón permite tener clases fáciles de mantener, entender y reutilizar. La utilización de comentarios en cada método, el uso de rutinas sencillas de entender y los nombres sugerentes son muestras del uso de este patrón en la solución.

Se puede observar dicho patrón en la clase *PostgreSQL* encargada de establecer las conexiones a las Bases de datos, para lo cual instancia la clase *Pycog\_2* que realiza las consultas y las iteraciones dentro de las Bases de datos.

**Bajo acoplamiento:** permite soportar bajas dependencias, un bajo impacto del cambio e incremento de la reutilización. Este patrón se tiene en cuenta por la importancia de realizar un diseño de clases independiente que puedan soportar los cambios de una manera fácil y permitan la reutilización. El uso de los patrones Experto y Creador favorecen al bajo acoplamiento entre las clases de la Herramienta Informática y el mantenimiento del mismo.

Este patrón se puede evidenciar en los ejemplos antes mencionados.

**Controlador:** asigna la responsabilidad de controlar el flujo de eventos de la Herramienta Informática, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes de la herramienta.

El uso de este patrón se ejemplifica en la clase *Controladora\_Vista\_Usuario* la cual asigna las responsabilidades de controlar los eventos de autenticar usuario, crear proceso y difundir resultados a las clases *Controladora\_Autenticar*, *CrearProceso* y *Difundir\_Resultados* respectivamente.

### 3.4. Estructura del proyecto

Los archivos del proyecto son almacenados en una estructura jerárquica, de tipo árbol. El proyecto puede estar compuesto por una o más aplicaciones las cuales se dividen en los módulos definidos anteriormente para el desarrollo del proceso KDD. Dicha estructura se muestra acto seguido:



Figura 3.2. Estructura del proyecto

### 3.5. Tarjetas Cargo o clase, Responsabilidad y Colaboración (CRC)

A continuación se muestra un ejemplo de tarjeta CRC que representa la clase *CrearXMLMineria* la cual construye todos los pasos que contiene el proceso de aplicar minería, las restantes se pueden observar en el [Anexo 2](#):

Clase: CrearXMLMineria	
Responsabilidades	Colaboradores
<ul style="list-style-type: none"> <li>def __init__(self)</li> <li>def EntradaBdStep(self,nombre,host,consulta,usuario,passw)</li> <li>def NominalToBinominalStep(self,nombre)</li> <li>def FPGrowthStep(self,nombre,soporte, find_min_number_of_itemsets ,max_items,max_number_of_retries,</li> </ul>	---

## Capítulo 3. Diseño de la herramienta informática propuesta

<pre>min_number_of_itemsets) • def PrepararTagsConnect(self,diccOrden) • def CreateRulesStep(self, nombre,min_confidence,gain_theta,laplace_k, criterion,min_criterion_value) • def Write_as_TextStep (self,nombre,direccion)</pre>	
---	--

Tabla 3.1. Tarjeta CRC de la clase CrearXMLMineria

### 3.6. Diseño de la base de datos

Se definió para el desarrollo de HADOC una tabla que representa la entidad estudiante donde refleja su nombre, el rol asignado, así como las asignaturas correspondientes al departamento de ISW impartidas hasta su quinto semestre. Se muestra además, la entidad usuario que constituye los usuarios autenticados de los cuales se conoce nombre de usuario, contraseña y el rol que desempeña para el acceso a las funcionalidades de HADOC. A continuación se muestran dichas entidades en la figura 3.3:

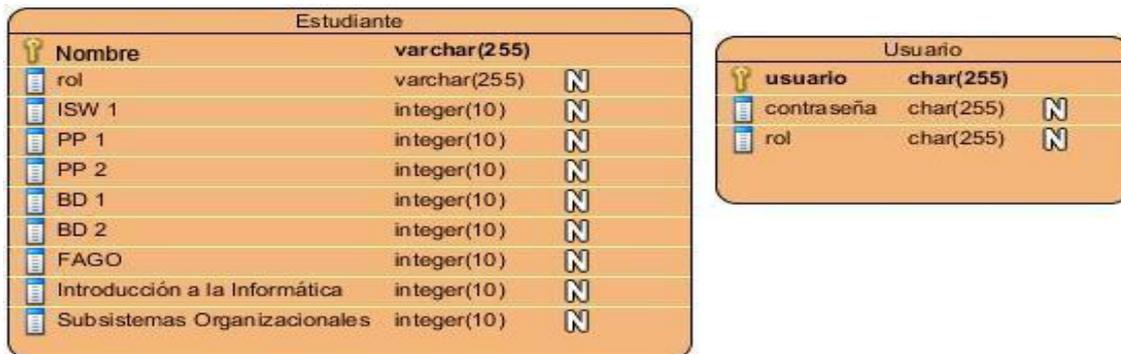


Figura 3.3. Entidades Estudiante y Usuario

### 3.7. Conclusiones del capítulo

- En este capítulo se presentó la arquitectura de tres niveles para estructurar la Herramienta Informática de forma tal que permita una adecuada organización para su implementación.
- Se expusieron los patrones de asignación de responsabilidades y de diseño utilizados con el objetivo de brindar una programación organizada al programador.
- Se mostraron los datos correspondientes a las entidades Estudiante y Usuario necesarios para realizar las funcionalidades de la Herramienta Informática.

# Capítulo 4. Implementación y Pruebas

## 4.1. Introducción

En este capítulo se muestran las fases de implementación y pruebas que propone la metodología utilizada. Primero se efectúa la implementación de las HU que fueron seleccionadas para cada iteración, luego se crean las tareas de programación para organizar la implementación de las HU y se realizan las pruebas seleccionadas.

## 4.2. Fase Implementación

En esta fase se muestran las tareas de programación asignadas a cada HU.

### 4.2.1. Tareas para cada HU

Durante cada iteración se presentan las tareas de programación que se desarrollan según la planificación antes propuesta.

### 4.2.2. Iteración 1

En esta iteración se abordaron las tareas referentes a las historias de usuarios que permiten una rápida retroalimentación con el cliente.

Tareas Críticas	Estimación(días)	Real(días)
Implementar las funcionalidades que permitan autenticar a los usuarios autorizados.	5/5	5/5
Implementar las funcionalidades que permitan adicionar usuarios para que accedan a la Herramienta Informática.	10/5	10/5
Implementar las funcionalidades que permitan modificar los datos de los usuarios con acceso a la Herramienta Informática.	10/5	10/5
Implementar las funcionalidades que permitan eliminar usuarios con acceso a la Herramienta Informática.	10/5	10/5

## Capítulo 4. Implementación y Pruebas

Implementar las funcionalidades que permitan almacenar la configuración de las conexiones a las bases de datos.	15/5	15/5
Implementar las funcionalidades que permitan cargar la configuración de las conexiones.	15/5	15/5
Implementar las funcionalidades que permitan preparar datos.	15/5	15/5
<b>Total</b>	80/5 días=16 semanas	80/5 días=16 semanas

Tabla 4.1. Tiempo de las tareas arribadas en la iteración 1

### 4.2.2.1. Tareas abordadas en la iteración 1

Tarea de Ingeniería	
<b>Número de la tarea:</b> 1	<b>Número de la Historia de Usuario:</b> 1
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan autenticar a los usuarios autorizados.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1
<b>Programador responsable:</b> Yankiel Belén, Rutminy Más	
<b>Descripción:</b> permite al usuario autorizado el acceso pleno a las funcionalidades de HADOC	

Tabla 4.2. Tarea de la historia de usuario: Autenticar usuario

Tarea de Ingeniería	
<b>Número de la tarea:</b> 2	<b>Número de la Historia de Usuario:</b> 2
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan adicionar usuarios.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 2
<b>Programador responsable:</b> Yankiel Belén, Rutminy Más	
<b>Descripción:</b> permite adicionar usuarios para que tengan acceso a HADOC.	

Tabla 4.3. Tarea de la historia de usuario: Adicionar usuario

Tarea de Ingeniería	
<b>Número de la tarea:</b> 3	<b>Número de la Historia de Usuario:</b> 3
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan modificar los datos de los usuarios con acceso a la Herramienta Informática.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 2

## Capítulo 4. Implementación y Pruebas

**Programador responsable:** Yankiel Belén, Rutminy Más

**Descripción:** permite modificar los datos de los usuarios que tienen acceso a HADOC.

Tabla 4.4. Tarea de la historia de usuario: **Modificar usuario**

Tarea de Ingeniería	
<b>Número de la tarea:</b> 4	<b>Número de la Historia de Usuario:</b> 4
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan eliminar usuarios con acceso a la Herramienta Informática.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 2
<b>Programador responsable:</b> Yankiel Belén, Rutminy Más	
<b>Descripción:</b> permite eliminar usuarios que tienen acceso a las funcionalidades de la HADOC	

Tabla 4.5. Tarea de la historia de usuario: **Eliminar usuario**

Tarea de Ingeniería	
<b>Número de la tarea:</b> 5	<b>Número de la Historia de Usuario:</b> 5
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan almacenar la configuración de las conexiones a las bases de datos.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 3
<b>Programador responsable:</b> Yankiel Belén, Rutminy Más	
<b>Descripción:</b> permite almacenar la configuración de las conexiones a las bases de datos y al servidor de correo.	

Tabla 4.6. Tarea de la historia de usuario: **Almacenar configuración de conexiones**

Tarea de Ingeniería	
<b>Número de la tarea:</b> 6	<b>Número de la Historia de Usuario:</b> 6
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan cargar la configuración de las conexiones.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 3
<b>Programador responsable:</b> Yankiel Belén, Rutminy Más	
<b>Descripción:</b> permite cargar la configuración de las conexiones a las bases de datos almacenadas anteriormente.	

Tabla 4.7. Tarea de la historia de usuario: **Cargar configuración de conexiones**

## Capítulo 4. Implementación y Pruebas

Tarea de Ingeniería	
Número de la tarea: 7	Número de la Historia de Usuario: 7
Nombre de la tarea: Implementar las funcionalidades que permitan preparar datos.	
Tipo de tarea: desarrollo	Puntos estimados: 3
Programador responsable: Yankiel Belén, Rutminy Más	
Descripción: permite obtener y filtrar de la BD de Gestión Universitaria las asignaturas del departamento de ISW impartidas a los estudiantes hasta el quinto semestre, así como el rol asignado de estos estudiantes en GESPRO. Estos datos son almacenados en la BD de HADOC.	

Tabla 4.8. Tarea de la historia de usuario: Preparar datos

### 4.2.3. Iteración 2

Durante la presente iteración se elaboró la tarea de implementación para aplicar la Minería de datos obteniendo Reglas de asociación.

Tareas Medias	Estimación(días)	Real(días)
Implementar las funcionalidades que permitan aplicar la Minería de datos.	15/5	15/5
<b>Total</b>	15/5días=3 semanas	15/5días=3 semanas

Tabla 4.9. Tiempo de las tareas arribadas en la iteración 2

#### 4.2.3.1. Tarea abordada en la iteración 2

Tarea de Ingeniería	
Número de la tarea: 8	Número de la Historia de Usuario: 8
Nombre de la tarea: Implementar las funcionalidades que permitan aplicar la Minería de datos.	
Tipo de tarea: desarrollo	Puntos estimados: 3
Programador responsable: Yankiel Belén, Rutminy Más	
Descripción: permite obtener Reglas de asociación de las opciones: asignaturas o asignaturas y roles, seleccionadas por el usuario.	

Tabla 4.10. Tarea de la historia de usuario: Aplicar minería

### 4.2.4. Iteración 3

Durante esta iteración se implementaron las funcionalidades que permitieron difundir y almacenar los resultados obtenidos tras aplicar la Minería de datos.

Tareas Bajas	Estimación(días)	Real(días)
Implementar las funcionalidades que permitan difundir resultados.	15/5	15/5
Implementar las funcionalidades que permitan generar reportes en formato <i>pdf</i> .	10/5	10/5
<b>Total</b>	25/5=5 semanas	25/5=5 semanas

Tabla 4.11. Tiempo de las tareas arribadas en la iteración 3

#### 4.2.4.1. Tareas abordadas en la iteración 3

Tarea de Ingeniería	
<b>Número de la tarea:</b> 9	<b>Número de la Historia de Usuario:</b> 9
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan difundir los resultados.	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 3
<b>Programador responsable:</b> Yankiel Belén, Rutminy Más	
<b>Descripción:</b> permite enviar el reporte de los resultados en formato <i>pdf</i> de forma opcional en un mensaje de correo electrónico a los interesados.	

Tabla 4.12. Tarea de la historia de usuario: Difundir conocimiento

Tarea de Ingeniería	
<b>Número de la tarea:</b> 10	<b>Número de la Historia de Usuario:</b> 10
<b>Nombre de la tarea:</b> Implementar las funcionalidades que permitan generar reportes en formato <i>pdf</i> .	
<b>Tipo de tarea:</b> desarrollo	<b>Puntos estimados:</b> 1
<b>Programador responsable:</b> Yankiel Belén, Rutminy Más	
<b>Descripción:</b> permite generar reporte en formato <i>pdf</i> de los resultados obtenidos al aplicar minería.	

Tabla 4.13. Tarea de la historia de usuario: Generar reportes en *pdf*

### 4.3. Fase Prueba

Las pruebas de la herramienta informática propuesta se dividieron en dos grupos: las pruebas unitarias planteadas por los programadores y las pruebas de aceptación diseñadas por el cliente.

#### 4.3.1. Pruebas unitarias

A continuación se muestra un ejemplo de prueba realizada a la funcionalidad Aplicar minería utilizando *pyunit*.

Primeramente se realiza la clase de pruebas unitarias importando la librería *unittest*:

```
import unittest
from ProcesoKDD.Interfaces.QTPY.Controladora_Vista_Usuario import Controladora_Vista_Usuario
```

Acto seguido se ejecuta Aplicar minería seleccionando todas las asignaturas expuestas:

```
class EjemploFixture(unittest.TestCase):
    def setUp(self):
        print 'Preparando contexto'
    #!SELECT "fago_1", "is 1", "pp 1", "pp 2", "sbd", "sbd 1", "sbd 2", "ii", "ae", "cf" FROM "public"."estudiante'
        self.proceso = Controladora_Vista_Usuario('', '')

    def test_RealizarMineria(self):
        r = self.proceso.RealizarMineria()
        self.assertEqual(r, 'terminado')
```

Finalmente se consigue que dicha funcionalidad esté desarrollada en un tiempo de ejecución que se muestra en la imagen siguiente:

```
-----
Ran 1 test in 46.293s

OK
```

Estos procedimientos se mostraron de igual manera al ejecutar la misma funcionalidad seleccionando todas las asignaturas y roles:

```
import unittest
from ProcesoKDD.Interfaces.QTPY.Controladora_Vista_Usuario import Controladora_Vista_Usuario
```

```
class EjemploFixture(unittest.TestCase):
    def setUp(self):
        print 'Preparando contexto'
        #SELECT "fago_1", "is 1", "pp 1", "pp 2", "sbd", "sbd 1", "sbd 2", "ii", "ae", "cf" FROM "public"."estudiante"
        self.proceso = Controladora_Vista_Usuario('', '')

    def test_RealizarMineria(self):
        r = self.proceso.RealizarMineria()
        self.assertEqual(r, 'terminado')
```

-----  
Ran 1 test in 46.515s

OK

### 4.3.2. Pruebas de aceptación

En la siguiente tabla se muestra la prueba de aceptación realizada a la funcionalidad Aplicar Minería de HADOC, las demás pruebas realizadas se pueden observar en el [Anexo 3](#):

Caso de prueba de aceptación	
<b>Código:</b> HU08-P01	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Aplicar Minería.	
<b>Descripción:</b> prueba para la funcionalidad de aplicar la Minería de datos.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>El usuario debe tener acceso a la Herramienta Informática.</li> </ul>	
<b>Entrada/Pasos de ejecución:</b>	
<ul style="list-style-type: none"> <li>El usuario selecciona los datos que desea minar de la vista minable Aplicar minería.</li> </ul>	
<b>Resultados esperados:</b> se realizan las Reglas de asociación.	
<b>Evaluación de la Prueba:</b> satisfactoria	

Tabla 4.14. **Caso de prueba de aceptación HU08-P01**

Las pruebas de aceptación se realizaron en tres iteraciones obteniendo en la primera iteración 4 no conformidades (NC), en la segunda se solucionaron 2 quedando 2 NC y finalmente en la tercera iteración se solucionaron las NC de la iteración anterior.

### **4.4. Conclusiones del capítulo**

- Se desarrollaron las tareas de ingeniería correspondientes a cada una de las HU de las diferentes iteraciones, además se explicaron las funciones de cada iteración con el objetivo de que el usuario pueda entender con facilidad la estructura de la implementación.
- Se realizaron las pruebas unitarias y de aceptación para verificar el correcto funcionamiento de la Herramienta Informática, obteniendo resultados satisfactorios en cada funcionalidad y solucionando las NC encontradas.

## *Conclusiones Generales*

El presente trabajo se culmina logrando cumplir los objetivos trazados utilizando las técnicas y metodologías escogidas para cada uno de estos, obteniendo los siguientes resultados:

- Se definió el marco teórico de la investigación donde se decidió utilizar la tarea descriptiva Reglas de asociación y el algoritmo FP-Growth además de las herramientas RapidMiner, Pentaho y la plataforma de desarrollo Eclipse.
- Se realizó un análisis sobre las metodologías de desarrollo de software donde se seleccionó la metodología XP como la más apropiada.
- Se analizaron diferentes trabajos que aplican la Minería de datos en el entorno educativo llegando a la conclusión de que ninguno de ellos brinda una solución a la problemática expuesta.
- Se diseñó un modelo de solución de la Herramienta Informática propuesta que permite la realización de los procesos de Preparación de datos, Minería de datos y Difusión de resultados y se implementaron dichos módulos.
- Se validó la Herramienta HADOC a partir de pruebas unitarias y de aceptación comprobándose la efectividad del código y la satisfacción del cliente con las funcionalidades presentadas.

## *Recomendaciones*

Se recomienda, para futuras investigaciones:

- Continuar recolectando datos de los estudiantes en función de seguir fortaleciendo el conocimiento de la organización y potenciar el desarrollo de investigaciones más reveladoras y generalizables.
- Incorporar otras técnicas de minería de datos a la propuesta realizada en esta investigación.

## *Referencias bibliográficas*

1. Acosta, R. V. (2007). Obtención de patrones y reglas en el Sistema Docente del Instituto Superior Politécnico José A.Echeverría (CUJAE) utilizando Minería de Datos. C. La Habana
2. Alcover, J. B., P Blesa, MA Calduch (2007). Predicción del rendimiento académico de alumnos de primer año de la FACENA (UNNE) en función de su caracterización socioeducativa.
3. Alonso, C. (2013). "Reglas de Asociación".
4. Amil, M. L. C. (2010). Instrumentos Estadísticos de Investigación Económica. .
5. Benavides, S. (2007). "Aplicación web que utiliza Minería de datos para generar conocimiento".
6. Bustamante, F. I. B. (2010). Ingeniería en computación, especialización sistemas tecnológicos.
7. Cabrera, A. S., Walfrido; Martínez Leye, Olga; Mompié, Iyanet. (2012). "Revista Iberoamericana de Evaluación Educativa", from [http://www.rinace.net/riee/numeros/vol5-num2/art11\\_hm.html](http://www.rinace.net/riee/numeros/vol5-num2/art11_hm.html).
8. Cardona, J. C. G. (2010). Sistema de apoyo para la acreditación de la calidad de programas académicos de la Universidad de Caldas, aplicando técnicas en Minería de Datos.
9. Consultores, W. (2011). "KDD: Proceso de Extracción de conocimiento".
10. Córcoles, J. E. (2011). "Requisitos No Funcionales ", from <http://prezi.com/30ymlpvpqvch/requisitos-no-funcionales/>.
11. Chapman, P. (2012). "Metodología CRISP-DM para minería de datos".
12. Deseta, L. (2013). "Desarrollo ágil de software." 2013, from [http://www.dosideas.com/wiki/Desarrollo\\_Agil\\_De\\_Software](http://www.dosideas.com/wiki/Desarrollo_Agil_De_Software).
13. Despaigne, H. H., Julio Antonio (2011). HERMINWEB, Herramienta informática de Minería de Uso de la Web sobre los registros de navegación por Internet. .
14. EcuRed (2012). Eclipse, entorno de desarrollo integrado.
15. EcuRed (2013). Metodologías Tradicionales.
16. Espinosa, R. (2010). Teoría de Data Mining. DATA PRIX.
17. Esquivés, A. A. (2011). PyQt + QtDesigner.
18. Fonseca, T. (2011). "Minería de datos." from <http://hdl.handle.net/2445/19862>.

19. Freddy, J. V. (2012). "Metodología XP", from <http://ingsoftware072301.obolog.com/metodologia-xp-2012877>.
20. González, F. A. (2012) "Minería de Datos 2012 II".
21. ISPJAE, F. d. A. (2008). "Arquitectura y urbanismo, Volúmenes 3-4."
22. Larman, C. (2008). "UML y Patrones. Introducción al análisis y diseño orientado a objetos."
23. Larose, D. T. (2009). "Data Mining. Methods and Models. Department of Mathematical Sciences. Central Connecticut State University."
24. Lent, D. W. (2010). Ateneo - Universidad Nacional Mayor de San Marcos.
25. Letelier, P. (2011).
26. Leyva, Y. O. (2011). Procesamiento distribuido de los registros de navegación de Internet.
27. Limonta, Y. P. (2007). Implementación de algoritmo para la generación de reglas de asociación representativas, UCI.
28. López, F. (2013). "Metodologías Ágiles." from <http://interfaces-siges.googlecode.com/files/Metodologias%20Agiles.doc>.
29. López, J. M. M. (2006). Técnicas de Análisis de Datos.
30. Marcé, A. C. C. (2011). Sistema Distribuido Cliente-Servidor. La Habana, UCI.
31. Marín, C. (2009). Fases de un proyecto de Minería de datos.
32. Marín, C. (2009). "Minería de Datos.", from <http://mineriadatos.com>.
33. Matías, Y. E. L., Katia (2010). Propuesta de Módulo de Procesamiento Inteligente de Datos para el sistema Airesweb. Habana, UCI.
34. Moine, J. M. (2010). "Estudio comparativo de metodologías para minería de datos."
35. Molteni, M. B. (2007). Detección de Patrones de Producción Educativa Basada en Minería de Datos.
36. Montalvo, M. M., Vanessa; André, Margarita; Baldoquín, María Gulnara (2013). "Gestión del conocimiento para la obtención de patrones que contribuyan a la formación de equipos de proyectos de software."
37. Muñoz, P. L. (2012). "¿Qué es un proceso de desarrollo de software?".
38. Pentaho, C. (2010). from <http://www.pentaho.com/>.
39. Pérez, S. (2007). Obtención de Reglas y Patrones en el Proceso Académico de la Universidad de Ciencias Informáticas.

40. Pinho, J. (2010). Métodos de clasificación basados en asociación aplicados a sistemas de recomendación. Departamento de Informática y Automática. Salamanca, Universidad de Salamanca.
41. Porcel, E. A. (2010). Revista electrónica de investigación educativa, versión On-line ISSN 1607-4041. Ensenada, Baja California, México.
42. QtProject. (2013). from <http://doc.qt.digia.com>.
43. Quintales, L. A. M. (2013). "Aplicación de técnicas de Minería de Datos en la construcción y validación de modelos predictivos y asociativos a partir de especificaciones de requisitos de software."
44. Ramírez, I. D. P. (2008). "Metodologías ágiles. ¿Cómo desarrollo utilizando XP?", from <http://ccia.cujae.edu.cu/index.php/siia/siia2008/paper/download/.../246>.
45. Ricardo, J. A. H. (2011). Sistema de servicio de SMS basado en Gammu (JAMS). La Habana, UCI.
46. Ril, E. B. (2012). Descubrimiento de conocimiento a partir de lecciones aprendidas documentadas en los procesos de cierre de proyectos informáticos. Habana, UCI.
47. Sarasa, R. B. (2008). "Minería de Datos aplicada a la Gestión Docente del Instituto Superior Politécnico José Antonio Echeverría".
48. Silvana, M. (2009). Minería de Datos Secuenciales.
49. Suárez, M. C. (2009). "Teleidentificador Personal: Diseño de la arquitectura de la plataforma manejadora de peticiones".
50. Timarán, R. (2010) "Una lectura sobre deserción universitaria en estudiantes de pregrado desde la perspectiva de la minería de datos." Revista Científica Guillermo de Ockham.
51. Timarán, R. (2013). "Análisis de desempeño de EquipAsso: Un algoritmo para el cálculo de Itemsets frecuentes basado en operadores algebraicos relacionales.", 2013, from [http://www.academia.edu/1596194/Analisis\\_de\\_desempeno\\_de\\_EquipAsso\\_Un\\_algoritmo\\_para\\_el\\_calculo\\_de\\_Itemsets\\_frecuentes\\_basado\\_en\\_operadores\\_algebraicos\\_relacionales](http://www.academia.edu/1596194/Analisis_de_desempeno_de_EquipAsso_Un_algoritmo_para_el_calculo_de_Itemsets_frecuentes_basado_en_operadores_algebraicos_relacionales).
52. Tuya, J. R., Isabel (2007) "Técnicas cuantitativas para la gestión de ingeniería de software".
53. Ucan, S. (2011). Aplicaciones exitosas de la Minería de Datos.

54. Uñoja, R. H. (2012). Metodologías de desarrollo de software Tradicionales vs Ágiles. Blogger. Bolivia.
55. Ventura, S. (2013). "Usando minería de datos para la continua mejora de cursos de e-learning."
56. Vigo, F. B. U. (2012). "III Jornada Sage/Python." from <http://webs.uvigo.es/sage2012/>.
57. Villavicencio, A. (2012). Búsqueda de patrones frecuentes en un conjunto de datos mediante el algoritmo FPGrowth. Universidad de Chile.
58. Wilfrod, I. (2013). "Minería de datos: herramienta de apoyo en la selección de equipos de proyectos informáticos".