VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

MAESTRÍA EN GESTIÓN DE LA INFORMACIÓN E INTELIGENCIA DE NEGOCIOS

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE MASTER EN GESTIÓN DE LA INFORMACIÓN E INTELIGENCIA DE NEGOCIOS

PLANTEAMIENTO DE UN MODELO DE ESTIMACIÓN DE PROYECTOS DE SOFTWARE EN ETAPAS TEMPRANAS BASADO EN REDES NEURONALES ARTIFICIALES

AUTOR: CORAL CORAL, HENRY RAMIRO

DIRECTOR: FONSECA CARRERA, EFRAÍN RODRIGO

SANGOLQUÍ

2018

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MAESTRÍA EN GESTIÓN DE LA INFORMACIÓN E INTELIGENCIA DE NEGOCIOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, **"PLANTEAMIENTO DE UN MODELO DE ESTIMACIÓN DE PROYECTOS DE SOFTWARE EN ETAPAS TEMPRANAS BASADO EN REDES NEURONALES ARTIFICIALES"**, realizado por el señor Ing. **HENRY RAMIRO CORAL CORAL**, ha sido analizado por el software antiplagio y revisado en su totalidad, determinándose que cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, por lo tanto autorizo para ser sustentado públicamente.

SANGOLQUÍ, 28 de febrero de 2018.

Ing. Rodrigo Fonseca, PHD

DIRECTOR DE TESIS.

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MAESTRÍA EN GESTIÓN DE LA INFORMACIÓN E INTELIGENCIA DE NEGOCIOS

AUTORÍA DE RESPONSABILIDAD

Yo, **HENRY RAMIRO CORAL CORAL**, con cédula de identidad Nª C.C. 1714864830, declaro que el trabajo de titulación **"PLANTEAMIENTO DE UN MODELO DE ESTIMACIÓN DE PROYECTOS DE SOFTWARE EN ETAPAS TEMPRANAS BASADO EN REDES NEURONALES ARTIFICIALES"**, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros, que constan en las citas bibliográficas.

Consecuentemente, declaro que este trabajo es de mi autoría, en virtud de ello soy responsable del contenido, veracidad y alcance de la investigación mencionada.

SANGOLQUÍ, 28 de febrero de 2018.

HENRY RAMIRO CORAL CORAL
C.C. 1714864830

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MAESTRÍA EN GESTIÓN DE LA INFORMACIÓN E INTELIGENCIA DE NEGOCIOS

AUTORIZACIÓN

Yo, **HENRY RAMIRO CORAL CORAL**, autorizo a la Universidad de las Fuerzas Armadas ESPE, publicar en la biblioteca virtual de la institución el presente trabajo de titulación **"PLANTEAMIENTO DE UN MODELO DE ESTIMACIÓN DE PROYECTOS DE SOFTWARE EN ETAPAS TEMPRANAS BASADO EN REDES NEURONALES ARTIFICIALES"**, cuyo contenido ideas y criterios son de mi autoría y responsabilidad.

SANGOLQUÍ, 28 de febrero de 2018.

HENRY RAMIRO CORAL CORAL
C.C. 1714864830

DEDICATORIA

Este trabajo está dedicado a:

Matías Sebastián, Gloria Esperanza y Andrea Estefanía; representantes del amor en mi vida y de darle significado a la misma.

AGRADECIMIENTO

Quiero agradecer a las siguientes personas; que sin su ayuda no habría sido posible realizar la investigación; redactar este documento ni llegar a exponerlo el día de hoy.

Mis más sinceros agradecimientos a:

- Ing. Dennys Guzmán, Gerente de Sistemas en Gestor y el experto en SQL que me ayudo en el proceso de exploración y descubrimiento de los datos.
- Ing. Rodrigo Fonseca, el director de la tesis.
- Ing. Tatiana Gualotuña, oponente del proceso
- Ing. Germán Ñacato, director de la maestría.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

La estimación de tiempos en etapas tempranas de un proyecto de software se basa principalmente en el uso de la técnica de Juicio Experto, lo cual muchas veces y dependiendo de la complejidad del proyecto puede ocasionar graves problemas y muchas veces el fracaso del mismo. Esta problemática se presenta en la mayoría de empresas de software, ya que la estimación de tiempos para un proyecto se la realiza en la etapa de licitación basado en un documento de alto nivel que detalla el alcance esperado y sirve como base para definir la duración y costos del mismo; valores que no pueden ser modificados una vez que se ha levantado los requisitos específicos de las funcionalidades que contempla el proyecto y se conoce el esfuerzo real que debe ser realizado. Esta investigación propone el uso de la información histórica que poseen las empresas desarrolladoras de software de sus proyectos pasados para proponer un modelo de estimación basado en minería de datos, el cual utiliza redes neuronales artificiales para calcular la estimación de nuevos requerimientos en base a un número determinado de parámetros como el número de entidades, el número de reglas de negocio y la dificultad de implementación para cada funcionalidad

PALABRAS CLAVES:

- **GESTIÓN DE PROYECTOS DE SOFTWARE.**
- **ESTIMACIÓN DEL ESFUERZO.**
- **HERRAMIENTAS DE ESTIMACIÓN.**
- **MINERÍA DE DATOS, REDES NEURONALES ARTIFICIALES**.

ABSTRACT

Time estimation at early phases of a software project is mainly based in the use of the Expert Judge technique; same technique that in many occasions, and depending of a project's complexity, could cause serious problems and ultimately might result in the failure of a project. This complication presents itself in most software companies, since the time estimation for a software project is done during the elicitation stage and it is described in a high level document that details the expected scope and serves as a base to define the duration and cost. However this values cannot be modified later, once the specific requirements of the functionalities contemplated in the project have been raised and the real effort that must be made is known. This research proposes the use of historical information held by software development companies of their past projects to develop an estimation model based on data mining. This model uses artificial neural networks to calculate the estimation of new requirements based on specific parameters such as the number of entities, the number of business rules and the difficulty of implementation for each functionality. The obtained results in the investigation when applying the developed information model were not completely satisfactory; although, they served as the basis for improving the process of recording information so that it can be used in future projects.

KEY WORDS:

- **SOFTWARE PROJECT MANAGEMENT.**
- **EFFORT ESTIMATION.**
- **ESTIMATION TOOLS.**
- **DATA MINING, ARTIFICIAL NEURAL NETWORKS**.

CAPÍTULO I
EL PROBLEMA

1.1 Antecedentes

La industria del software en el Ecuador ha crecido de forma considerable en los últimos años; sin embargo, comparada con otros países de américa latina se puede decir que todavía se encuentra en etapas iniciales de desarrollo (Vela Casado, 2010). De acuerdo al informe “La cadena del software en Ecuador: Diagnóstico, visión estratégica y lineamientos de política” (Calderón, Castillo, & Bercovich, 2013) “en el Ecuador existen cerca de 500 empresas desarrolladoras de software con un volumen de facturación anual de 500 millones de dólares y genera 7000 empleos directos”; comparando estas cifras con las del año 2010 según un informe de la Asociación Ecuatoriana de Software, AESOFT, se registra la existencia de 416 empresas que se dedican a actividades en el sector del Software, cuya facturación asciende a US$242 millones de dólares, lo cual representa el 0,4% del PIB, el 2.1% de los ingresos totales del Gobierno Central y el 3.3% de los ingresos tributarios (Melany Gualavisí M., 2011), por lo que se puede apreciar que el sector ha experimentado un crecimiento de casi el doble en tres años.

Actualmente, las micro, pequeñas, medianas y grandes empresas dedicadas al desarrollo de la industria del software en el Ecuador han diversificado su cartera de productos y/o servicios; esta diversificación se ve reflejada en la constitución de diferentes tipos de empresas como: empresas de servicios (administración, operación y capacitación), de consultoría (integración, implementación, planeación), de productos estandarizados (software empaquetado, bajo licenciamiento), y de desarrollo y adecuación de software (bajo pedido) (Melany Gualavisí M., 2011). Es importante señalar que las grandes empresas de software asentadas en el país han optado, en su mayoría, por la prestación de servicios, mientras que el resto de empresas se han inclinado por la elaboración de productos estandarizados y el

desarrollo y adecuación del software, en tanto que las PYMES concentran su actividad en ventas de productos cibernéticos (Melany Gualavisí M., 2011).

En el mundo es un hecho que un porcentaje significativo de proyectos cuyo fin es el desarrollo de sistemas de software de gran tamaño y complejidad tiende a retrasarse, exceder el presupuesto de manera significativa o cancelarse por completo a la mitad del camino, debido a severas imprecisiones de estimación durante la fase de especificación de requisitos (Jones C., 2007). Lastimosamente muchas empresas en su afán de ganar una licitación en un mercado altamente competitivo optan por rebajar sus costos basados en estimaciones demasiado optimistas que a la final les han pasado factura (Vela Casado, 2010), en el mejor de los casos con demoras y mínimos márgenes de ganancia o con costos en contra y en el peor de los casos se ha producido hasta la quiebra de las empresas y el inicio de acciones legales a los accionistas de las mismas.

En el mercado ecuatoriano de acuerdo a un informe de la AESOFT, los principales clientes de las empresas desarrolladoras de software son las empresas privadas con el 78% de la totalidad de mercado y las empresas públicas con el 22% (AESOFT, 2016); el mercado es muy competitivo por lo que muchas empresas han optado por buscar clientes en el extranjero; los cuales les rinden mejores réditos económicos (Vela Casado, 2010).

El proceso de contratación de proyectos de desarrollo de software depende del sector contratante; de acuerdo a la ley de Contratación Pública del estado ecuatoriano, las empresas públicas deben exponer sus requerimientos en el portal de Compras Públicas; mientras que las empresas del sector privado tienen diferentes procesos para la contratación los cuales dependen de cada empresa. En una revisión a diferentes contratos suscritos por la empresa Gestor[1] con empresas públicas y privadas, ecuatorianas y de latinoamérica; se ha podido evidenciar la severidad en las

[1] Gestor es una empresa fundada en 1997 con experiencia en el desarrollo de soluciones tecnológicas especializadas para la Industria Financiera Internacional. http://www.gestorinc.com/ empresa/quienes-somos/

cláusulas de penalidades por retrazos y la imposibilidad de realizar alcances o modificaciones a los valores y montos especificados inicialmente en los contratos.

Las ofertas que realizan las empresas desarrolladoras de software se basan en la comprensión de *"Documentos de Términos de Referencia o Pliegos"* para las empresas públicas (Subsecretaría de Gestión Estratégica e Innovación, 2011) o *"Solicitud de Propuesta RFP"* para las empresas privadas (Humboldt State University, 2015) los cuales son documentos que detallan de forma general las necesidades de los interesados en contratar un desarrollo de software .

La mejor alternativa recomendada por los expertos en ingeniería de software para la estimación del esfuerzo, cuando la especificación de requerimientos no es clara y detallada es utilizar herramientas automatizadas (Jones C., 2007); sin embargo, su aplicación es casi nula, ya que la aplicación y uso de las mismas requiere de personal especializado y además implica una gran inversión económica que lamentablemente no se encuentra al alcance de pequeñas y medianas empresas (Jones C., 2007).

1.2 Tema

En función de la problemática indicada, se desea realizar el "*Planteamiento de un Modelo de estimación de Proyectos de Software en etapas tempranas basado en Redes Neuronales a partir de información histórica de los proyectos de implantación de la empresa Gestor*"; para mejorar la predicción de la estimación inicial del esfuerzo necesario para el desarrollo de futuros proyectos.

1.3 Planteamiento del Problema

1.3.1 Contextualización del Problema

La planificación de un proyecto de software en la etapa de "*elaboración de propuesta*" es meramente especulativa, ya que al basar la planificación y estimación en un documento de alto nivel, no se dispone de un conjunto completo de requisitos claramente detallados que especifiquen el producto que se va a construir, una parte

importante de la propuesta es la elaboración de la planificación del proyecto, el cual al no tener claro el alcance y todas las implicaciones del mismo, es elaborado de forma general por lo que en caso de ganar la licitación el mismo debe ser re-planteado (Pressman, 2010).

En la elaboración de la oferta para una licitación, se debe incluir el precio que se propondrá al cliente para desarrollar el software. Como punto de partida para calcular este precio, se elaborara una estimación de sus costos para completar el trabajo del proyecto (Jones C., 2007). La estimación implica calcular cuánto esfuerzo se requiere para completar cada actividad y, a partir de esto, calcular el costo total de las actividades, la descripción del proceso es simple y fácilmente realizable; sin embargo, el lograr estimar el esfuerzo real es una actividad que con el tiempo se ha convertido en un arte, antes que en un proceso de ingeniería. Son pocos los ingenieros de desarrollo y líderes de proyecto capaces de realizar una estimación que se acerque a la realidad. Una vez que tenga una estimación razonable de los costos probables, estará en condiciones de calcular el precio que cotizará al cliente (Pressman, 2010).

Revisando información de la industria de software en países desarrollados, se conoce que cerca del 80% de grandes corporaciones emplean herramientas automatizadas de estimación de software, en esta línea 30% utiliza dos o más de herramientas automatizadas de estimación a veces para el mismo proyecto; mientras otro 15% utiliza especialistas en estimación de costos. Para compañías pequeñas con menos de 100 empleados en el área de software sólo cerca del 25% utilizan herramientas automatizadas de estimación de software (Jones C., 2007).

El uso o desarrollo de herramientas de estimación de esfuerzo para etapas tempranas de un proyecto es viable cuando se cuenta con información histórica que sirva de insumo a la herramienta de estimación; para la elaboración de este trabajo de investigación se cuenta con información detallada sobre tiempos de desarrollo, aseguramiento de calidad y reprocesos (tiempo destinado a la corrección de errores) de más de 3500 requisitos desarrollados en la implantación del sistema “Gestor Fiducia Fondos” utilizado para la gestión de negocios fiduciarios de la emrpesa Gestor.

1.3.2 Análisis Crítico

Una vez que se ha contextualizado el problema que ha motivado la realización de esta investigación es necesario cuestionar su origen, causas, efectos y consecuencias; para este análisis crítico se ha utilizado la técnica de árbol de problemas.

¿Cómo mejorar el nivel de precisión en la estimación del esfuerzo en proyectos de desarrollo de software, en etapas tempranas del proyecto sin contar con el detalle completo de los requisitos del producto final?

Entre las causas del problema en base a la experiencia profesional del investigador y la investigación bibliográfica podemos mencionar las siguientes:

- Condiciones del mercado que obligan a las empresas de software a presentar propuestas sin contar con un alcance claro y detallado,
- Falta de registros históricos de tiempos estimados y reales en proyectos realizados anteriormente por las empresas (Jones C., 2007),
- Desconocimiento en la aplicación de técnicas y herramientas recomendadas por la Ingeniería de Software y la Gestión de proyectos que ayudan a disminuir el riesgo en el proceso de estimación en fases tempranas (Jones C., 2007).

Este análisis nos lleva a plantear las siguientes interrogantes:

- ¿Qué tan cierta y válida es la información recolectada en proyectos pasados?
- ¿Cómo se puede utilizar información recolectada de proyectos pasados para mejorar los procesos de estimación en futuros proyectos?
- ¿Qué grado de confianza puede dar el uso de un modelo de redes neuronales en la estimación del esfuerzo en la etapa de elaboración de la oferta de un desarrollo de software?

1.3.3 Prognosis

La contratación de servicios de software continuará realizándose en base a licitaciones públicas y con la calificación de ofertas económicas y técnicas presentadas por los proveedores en base a documentos de especificación de requisitos de alto nivel, es por esto que la industria del software debe mejorar sus procesos de estimación temprana; automatizándolos para disminuir al máximo el riesgo de fracaso de sus proyectos.

Si las empresas de software independientemente de su tamaño y experiencia continúan sin llevar registros detallados de sus tiempos empleados en tareas de planificación, análisis, diseño, codificación, pruebas, reprocesos y documentación; corren el riesgo de cometer errores que afecten su credibilidad técnica, salud económica y reputación en general; lo que podría causar en algunos casos la quiebra y desaparición de las mismas.

1.3.4 Formulación del Problema

¿Cómo mejorar la estimación del esfuerzo necesario en un proyecto de software en etapas tempranas mediante el uso de información histórica de proyectos similares?

1.3.5 Preguntas Directrices

- ¿Es válida y suficiente la información que la empresa Gestor dispone para la construcción de un modelo de estimación?
- ¿Cuáles son los factores que influyen en mayor grado al momento de estimar requisitos de software de alto nivel?
- ¿El modelo producto de la investigación podrá ser generalizado para su utilización en otros productos o proyectos de la empresa?
- ¿Cuál es la diferencia entre tiempos de esfuerzo estimados y tiempos reales en la ejecución de proyectos de desarrollo de software similares?

- ¿En qué porcentaje mejorará la estimación del esfuerzo en proyectos de software el uso de una herramienta automática implementada sobre la base de información histórica?

1.3.6 Delimitación del Problema

Delimitación Espacial del Problema

La presente investigación se realizará en base a la información histórica obtenida de los proyectos de implantación del producto “Gestor Fiducia Fondos” de la empresa Gestor que se encuentra ubicada en:

Provincia: Pichincha
Cantón: Quito
Parroquia: Benalcázar
Barrio: Bellavista

Delimitación Temporal del Problema

Para realizar la investigación y proponer un modelo de estimación del esfuerzo de proyectos de software en etapas tempranas se ha utilizado la información histórica de los proyectos de la empresa Gestor en un período comprendido entre los años 2010 al 2016.

Delimitación del Contenido del Problema

El resultado del proyecto de investigación es la definición de un modelo de minería de datos que en base a información histórica, permita mejorar la precisión de la estimación del esfuerzo en proyectos de desarrollo de software orientados al desarrollo y mantenimiento de nuevos requerimientos de un software empaquetado.

1.4 Justificación del Proyecto

Ente las razones que justifican el proyecto, podemos mencionar las siguientes:

- Las condiciones de los proyectos actuales que fuerzan a las empresas a presentar propuestas para proyectos de desarrollo de software sin información detallada; impide el uso de técnicas propias de la Ingeniería de Software para asegurar el éxito de los mismos; con la realización de esta investigación se busca logar un acercamiento entre los conceptos y técnicas que se imparten en la academia con la aplicación real en la industria de software local.

- Las pequeñas y medianas empresas de software realizan la estimación del esfuerzo en proyectos de software de forma manual y empírica, por lo que la definición de un modelo basado en información histórica del medio local ayudará a mejorar la precisión de sus estimaciones iniciales.

- En la industria local son pocas las empresas que llevan registros detallados sobre el trabajo realizado en sus proyectos de software, con el desarrollo de esta investigación se pretende crear conciencia en la importancia de la información histórica para la estimación de proyectos futuros.

- El modelo de estimación del esfuerzo, se lo realizará para proyectos de implantación y mantenimiento de software empaquetado; por lo que muchas empresas locales que se dedican a este giro de negocio podrán mejorar sus procesos al revisar los resultados de esta investigación.

1.5 Objetivos del Proyecto

1.5.1 Objetivo General

Definir un modelo de estimación del esfuerzo en etapas tempranas para Proyectos de Software basado en Redes Neuronales Artificiales.

1.5.2 Objetivos Específicos

- Utilizar la información historíca de los proyectos de implantación de la empresa Gestor en la construcción del modelo de estimación.

- Validar la calidad de la información que la empresa Gestor mantiene y verificar su uso en la construcción del modelo de estimación.

- Evaluar el nivel de fiabilidad del modelo de estimación planteado utilizando un subconjunto de los registros históricos que dispone la empresa Gestor.

CAPÍTULO II
MARCO TEÓRICO

2.1 Antecedentes Investigativos

La estimación del esfuerzo para proyectos de software es un tema que nunca llegará a ser exacto y cuya investigación seguirá desarrollándose en los próximos años. Cada año aparecen de nuevos lenguajes de programación, nuevos tipos de aplicaciones y nuevas plataformas de despliegue; muchas de estas nuevas tecnologías tienen éxito y logran permanecer en la cumbre por al menos 10 años; sin embargo, muchos son olvidados rápidamente o sustituidos por tecnologías similares (Pressman, 2010).

La mayoría de investigaciones que abordan la temática de la estimación del esfuerzo y proponen nuevos modelos y metodologías parten de los estudios realizados en la década de los 70, todos estos estudios están basados en el tamaño de la aplicación que va a ser desarrollada (Sommerville, 2011). La estimación del esfuerzo en base al tamaño medido en puntos de función ha evolucionado de forma constante desde su concepción, adaptándose a la evolución de los paradigmas de programación y de las metodologías para la elicitación de requerimientos (Jones C., 2007). Estas técnicas pueden ser utilizadas en proyectos donde se ha realizado la especificación de requisitos detallada y no ambigua; sin embargo, este no es el caso con el que se encuentran las empresas de software que deben realizar sus estimaciones en etapas tempranas (Varas, 2002).

De acuerdo a las publicaciones de (Jørgensen, 2004), el Juicio de Expertos es el método de estimación más difundido en la industria del software; además el autor observa que muchas veces la estimación de los expertos puede niveles de precisión similares o mejores a los producidos por métodos formales. En la investigación realizada por (Robiolo, Castillo, Rossi, & Santos, 2013) se compara metodos formales de estimación con el Juicio de un Experto, obteniendo mejores resultados con la estimación del mismo. Lastimosamente en la industria local, la falta de expertos

calificados en el mercado laboral ocasiona que no se puedan implementar modelos de estimación formales; las empresas pequeñas y medianas no cuentan con presupuesto para mantener de planta a estos expertos; la estimación es realizada por programadores de nivel senior y el líder de proyecto. Generalmente solo se asigna el 6% del presupuesto de un proyecto para la fase de planificación del mismo. Los métodos de estimación mayormente aplicados en la industria del software a nivel mundial son: Delphi y Regresión (Trendowicz, Münch, & Jeffery, State of the practice in software effort estimation: a survey and literature review, 2008).

Una vez que se ha determinado la imposibilidad del uso de las técnicas formales de estimación de esfuerzo en etapas tempranas de un proyecto se ha procedido a realizar una revisión de literatura en el que se utilice datos históricos de proyectos de software para la estimación del esfuerzo en etapas tempranas de un proyecto; tratando asi de ayudar a mejorar las estimaciones dadas por expertos en la industria local.

De la investigación realizada se puede mencionar los siguientes trabajos, (Pérez, González, Duque, Millane, & Ospina, 2006) presentan el desarrollo de un modelo dinámico para la estimación del esfuerzo en etapas tempranas basados en la empresa Orbitel S.A.; el modelo se basa en la selección de diferentes variables propias de la empresa y utilizan datos históricos de la misma; los resultados permiten definir la duración total de un proyecto. El trabajo de (Salvetto, Nogueira, & Segovia, 2004) se presentan modelos formales de estimación del tiempo y esfuerzo de desarrollo de sistemas de información utilizando cuatro parámetros de entrada como: eficiencia del grupo de desarrollo, volatilidad de los requerimientos, velocidad del desarrollo y la complejidad del sistema independiente de la tecnología del mismo; en este trabajo utilizan la herramienta Genexus como base para la obtención de información histórica que es la base en la construcción de sus modelos.

La necesidad de mejorar la precisión de las estimaciones ha despertado en la industria local la necesidad de mantener detalles del esfuerzo real utilizado en sus proyectos. Uno de los principales problemas de esta información es que solo se

refiere al tiempo empleado en cada una de las fases de desarrollo de un requerimiento es decir análisis, diseño, codificación, pruebas y documentación; sin embargo, las empresas no definen métricas que les permita evaluar los productos que están desarrollando (Luna, Álvarez, Espinaoza, Ambriz, & Nungaray, 2010).

La disposición de información histórica permite el uso de técnicas de aprendizaje automático y minería de datos para mejor el proceso de estimación. Las redes neuronales artificiales son capaces de encontrar patrones a partir de información que probablemente no sea fácil de encontrar utilizando métodos convencionales; es por esto que últimamente se están utilizando en muchas investigaciones relacionadas con la predicción del esfuerzo en proyectos de software (García, González, Colomo, López, & Ruiz, 2011). Un claro ejemplo del uso de redes neuronales artificiales en la estimación del esfuerzo y costo de proyectos de software se encuentra en el trabajo de (Almache, Raura, Ruiz, & Fonseca, 2015) en el cual se plantea un modelo de estimación tomando como entrada los atributos de la norma ISO/IEC 25000 de la calidad del software; con la aplicación de este modelo se obtiene las estimaciones de tiempo y costo para proyectos de software.

En la actualidad existen muchas publicaciones referentes al uso de redes neuronales artificiales para mejorar la precisión de la estimación; muchas comparan la eficiencia de algoritmos de aprendizaje automático con la información histórica que poseen (Quintero, Antón, Ferreira, & Gálvez, 2013); sin embargo, se debe tener en cuenta que el método de Juicio Experto en ciertos casos todavía puede resultar válido antes que el uso de otras técnicas automáticas (Robiolo, Castillo, Rossi, & Santos, 2013), las redes neuronales artificiales no son los únicos algoritmos que se utilizan en estas investigaciones (López & Dolado, 2007).

2.2 Red de Categorías

Para sustentar el contenido del marco teórico se ha estructurado una red de las principales categorías que intervienen en la explicación del tema de estudio; en la ***Figura. 1***, se muestra la red de categorías definida.

Figura. 1 Red de Categorías de las Variables de Investigación

2.3 Fundamentación Científica de la Variable Independiente

2.3.1 Proceso de Descubrimiento del Conocimiento en Bases de Datos

Como se mencionó en el apartado anterior la minería de datos es uno de los pasos de proceso de descubrimiento del conocimiento en bases de datos. Ahora se explicará en que consiste este proceso, junto con todas sus etapas.

KDD puede definirse como un proceso automático en el que se combina el descubrimiento y análisis en conjuntos de datos. El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice. Esta tarea implica generalmente pre-procesar los datos, hacer minería de datos y presentar resultados (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

El proceso KDD es interactivo e iterativo. Interactivo porque requiere de la intervención del investigador o analista de datos en cada fase e iterativo porque muchas veces puede ser necesaria la ejecución de varios ciclos, con el objetivo de obtener diferentes tipos de información. Las etapas del proceso son:

- Selección.
- Pre-procesamiento y limpieza.
- Transformación y reducción.
- Minería de datos.
- Interpretación y evaluación.

Antes de ejecutar cada una de estas etapas es necesario en primer lugar definir el proyecto de descubrimiento del conocimiento o también llamado proyecto de minería de datos. La definición debe basarse en las necesidades del usuario final y especificar de forma detallada el objetivo y metas del proyecto. La figura 3, muestra las fases del proceso KDD.

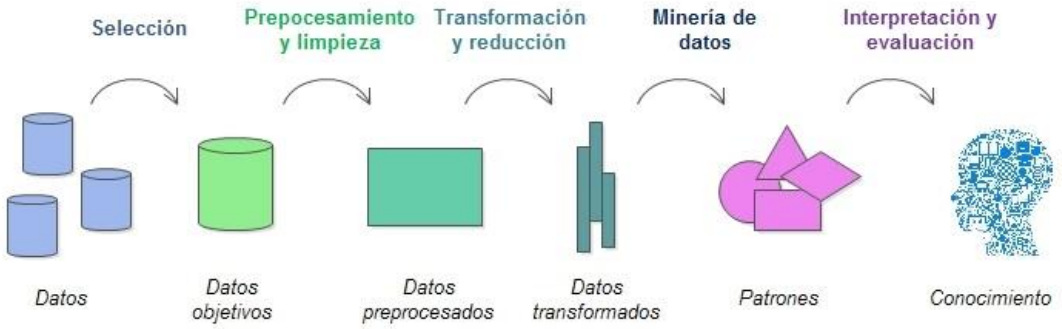

Figura. 2 Etapas del Proceso KDD

Etapa de Selección

Una vez que se ha definido el proyecto; el investigador crea un conjunto de datos objetivo en base a los datos almacenados en los repositorios de información de la empresa o cliente. El conjunto de datos puede tener todos los registros o una muestra representativa de los mismos; esto depende de la cantidad de información y de las capacidades de computacionales para su procesamiento (Timarán, Hernández, Caicedo, Hidalgo, & Alvarado, 2016).

Etapa de pre-procesamiento y limpieza

En esta etapa se analiza la calidad de los datos con los cuales se va a trabajar, es muy importante identificar datos que produzcan ruido para eliminarlos evitando así posibles complicaciones o errores al procesar los datos. Por lo general los datos que se remueven son los que tienen valores duplicados, nulos o inconsistentes; algunos datos pueden ser reemplazados utilizando para esto técnicas de estadística descriptiva como media, moda, mínimo y máximo. El proceso de limpieza de datos (data cleaning) debe realizarse en conjunto con el "dueño de los datos" es decir el usuario que tiene el conocimiento del negocio y puede dar sus apreciaciones sobre datos incorrectos, incompletos y fuera de límites (Timarán, Hernández, Caicedo, Hidalgo, & Alvarado, 2016).

Etapa de transformación y reducción

En esta etapa se buscan características útiles para representar los datos dependiendo de las metas definidas anteriormente. Se utilizan métodos de reducción de dimensiones o de transformación para disminuir el número efectivo de variables bajo consideración o para encontrar representaciones invariantes de los datos (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

Los métodos de reducción de dimensiones eliminan registros y columnas de las tablas. Los registros eliminados por lo general se refieren a datos idénticos no duplicados, es decir registros con los mismos valores pero generados en transacciones distintas; se debe tener en cuenta que no siempre se realiza esta eliminación ya que la presencia de datos idénticos puede tener un significado al momento de buscar patrones. Por otro lado la reducción de columnas, elimina columnas de las tablas que no son necesarias o que no contienen información de utilidad para el objetivo del proyecto (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

Etapa de Minería de Datos

El objetivo de la etapa minería de datos es la búsqueda y descubrimiento de patrones insospechados y de interés, aplicando tareas de descubrimiento como

clasificación, patrones secuenciales y asociaciones, entre otras. Dependiendo de las técnicas de minería de datos utilizadas se puede obtener modelos predictivos o descriptivos. Los modelos predictivos pretenden estimar valores futuros o desconocidos; mientras que los modelos descriptivos identifican patrones que explican o resumen los datos; estos sirven para explorar las propiedades de los datos examinados (Timarán, Hernández, Caicedo, Hidalgo, & Alvarado, 2016).

Etapa de interpretación y evaluación de datos

En esta etapa se interpretan los patrones descubiertos y en algunos casos se retorna a las anteriores etapas para la ejecución de nuevas iteraciones. Muchas veces esta etapa incluye la eliminación de patrones repetitivos, la traducción de patrones descriptivos o la visualización de los resultados de forma entendible para los usuarios finales. Antes de finalizar con la documentación de los resultados obtenidos se presenta la información a los usuarios finales para validarla y aclarar posibles dudas o mal entendidos (Timarán, Hernández, Caicedo, Hidalgo, & Alvarado, 2016).

2.3.2 Minería de Datos y Aprendizaje Automático

El avance tecnológico ha permitido disminuir los costos de los dispositivos de almacenamiento, cada vez nos preocupamos menos por la generación de información; la cual crece cada año en forma exponencial. Gracias a esta disminución de costos las bases de datos que almacenan la información de los sistemas transaccionales que utilizan a menudo las empresas también ha crecido; el crecimiento de las bases de datos está dado por dos factores: la cantidad de registros y el número de columnas de información de cada tabla. Gran parte de la información en las bases de datos se ha acumulado gracias a la facilidad de recopilación y almacenamiento que brinda la tecnología y no porque inicialmente haya sido considerada necesaria para su procesamiento en búsqueda de información relevante (Kenny, 2014).

Las empresas empezaron a crear almacenes de datos (datawarehouses) en los que inicialmente se utilizaron métodos basados en la estadística con el objetivo de encontrar información en base a hipótesis planteadas; sin embargo, gracias a investigaciones académicas surgió un nuevo campo de experimentación en el cuál se empezó a analizar todos los datos para buscar patrones que muestren información muchas veces inesperada u oculta (García & Molina, 2012).

Los seres humanos hemos buscado patrones desde los orígenes de nuestra especie; la observación permitió descubrir conocimiento que se encontraba en la naturaleza, este conocimiento permitió fue uno de los cimientos de nuestra evolución. En la actualidad los científicos dan sentido a los datos descubriendo patrones y formulando en teorías que pueden usarse para predecir lo que sucederá en situaciones nuevas (Witten, Frank, & Hall, 2011).

El uso de los datos para encontrar patrones que permitan clasificar a las personas por sus gustos, preferencias, ideología política, religión es cada vez más recurrente; gracias a las redes sociales, los sistemas de compras en línea y los buscadores de internet; que permiten la recolección de toda esta información de forma automática y sin que el usuario llegue a darse cuenta. El registro de esta información se ha dado especialmente en la última década, la cantidad de información generada ha imposibilitado el uso de almacenes de datos o bases de datos relacionales; por lo que han surgido nuevas formas de almacenamiento y procesamiento de la información (Kenny, 2014).

Según (Fayyad, Piatetsky-Shapiro, & Smyth, 1996) la minería de datos es: "Un proceso no trivial de identificación válida, novedosa, potencialmente útil y entendible de patrones comprensibles que se encuentran ocultos en los datos". El proceso debe ser automático o semiautomático. Los patrones descubiertos deben ser significativos, es decir deben representar alguna ventaja, generalmente económica.

Una definición más actualizada es la de (Kenny, 2014): "La minería de datos es un medio para producir información predictiva a partir de grandes cantidades de datos.

Es uno de los métodos de más rápido crecimiento de la predicción en el mundo de los negocios".

En el año 1996 con la presentación del artículo "The KDD Process for Extracting Useful Knowledge from Volumes of Data" de (Fayyad, Piatetsky-Shapiro, & Smyth, 1996) se popularizo el término "KDD" un acrónimo para "Descubrimiento del Conocimiento den Bases de Datos (Knowledge Discovery in Databases)". KDD se constituyó en el primer modelo aceptado por la comunidad científica que establece las etapas de un proceso de explotación de la información. En el modelo KDD la minería de datos es tan solo uno de los pasos del proceso. En la actualidad KDD y minería de datos, son utilizados indistintamente por en publicaciones científicas y literarias para hacer referencia al proceso completo de descubrimiento de información a partir de grandes volúmenes de datos (Moine, Haedo, & Gordillo, 2011).

Los patrones que busca la minería de datos, son utilizados para predecir el comportamiento de futuros datos; es por esto que es importante que los patrones encontrados puedan ser expresados como una estructura la cual pueda ser examinada, razonada y utilizada para la toma de decisiones. Estos patrones reciben el nombre de patrones estructurales, estos patrones pueden ser expresados como un conjunto de reglas, o árboles de decisión para que los investigadores puedan entenderlos y afinarlos de acuerdo a sus necesidades (Witten, Frank, & Hall, 2011).

Para comprender un poco más a la minería de datos, es necesario conocer sus raíces. Podemos decir que las dos raíces principales son la estadística y la inteligencia artificial, principalmente el área relacionada con el aprendizaje automático conocido popularmente como ("machine learning").

Para aclarar las diferencias entre la estadística y la inteligencia artificial podemos decir que, la segunda ha estado más preocupada en ofrecer soluciones algorítmicas con un costo computacional aceptable, mientras que la primera se ha preocupado más del poder de generalización de los resultados obtenidos (Aluja, 2001). Otra diferencia expresada por (Witten, Frank, & Hall, 2011), podría ser que la estadística se haya

preocupado más por probar hipótesis, mientras que el aprendizaje automático se ha preocupado más de formular el proceso de generalización como una búsqueda a través de posibles hipótesis; sin embargo esto es una simplificación excesiva ya que la estadística es mucho más que pruebas de hipótesis, y muchas técnicas de aprendizaje automático no implican ninguna búsqueda en absoluto. Adicionalmente se debe tener en cuenta que la mayoría de los algoritmos de aprendizaje automático utilizan pruebas estadísticas para validar sus procesos o minimizar los coeficientes de error (González, 2014).

Es importante puntualizar que en la actualidad los términos Minería de Datos y Aprendizaje Automático, se los utilizan indistintamente; si bien es cierto se podría decir que son similares; en realidad tienen su diferencia la cuál radica en el objetivo de cada una de las disciplinas. La minería de datos como se explicó anteriormente se encarga del descubrimiento de patrones desconocidos mientras que el aprendizaje automático utiliza y reproduce esos patrones para realizar predicciones (González, 2014).

Teniendo en cuenta el objetivo de este trabajo que es la determinación de un modelo que permita mejorar la precisión de la estimación del esfuerzo en proyectos de software en etapas tempranas; es que se utilizará la definición de técnicas de aprendizaje automático en lugar de técnicas de minería de datos; sin embargo, a continuación se presentará la dos clasificaciones en las que se puede observar la similitud entre ellas.

La clasificación de técnicas de minería de datos, divide a las mismas en dos grandes grupos: Técnicas de modelado dirigido por la teoría (técnicas predictivas) y Técnicas de modelado dirigido por los datos (técnicas descriptivas) (Pérez C. , 2004). Las técnicas predictivas se basan en un conocimiento teórico previo de los datos que van a ser analizados; en este caso el modelo obtenido debe ser contrastado para ser aceptado como válido. Las técnicas descriptivas suponen la no existencia de variables dependientes ni independientes; en este caso los modelos se crean automáticamente en base al reconocimiento de patrones (Pérez, 2014). La figura 4, muestra la clasificación de las técnicas de minería de datos.

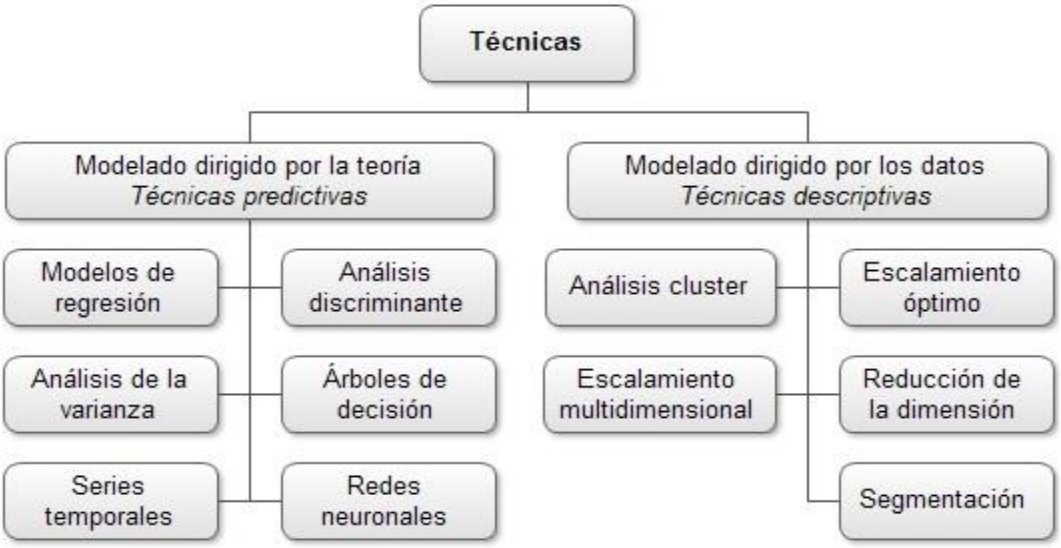

Figura 3 Técnicas de Minería de Datos

La clasificación de técnicas de aprendizaje automático, también divide a los algoritmos en dos grandes grupos de acuerdo a su tipo de aprendizaje: Algoritmos Supervisados y Algoritmos No Supervisados. Los algoritmos supervisados también conocidos como predictivos; predicen un valor desconocido en base a otros conocidos; mientras que los algoritmos no supervisados o descriptivos se encargan de descubrir patrones y asociaciones en los datos (García & Molina, 2012). La figura 5, muestra la clasificación de las técnicas de aprendizaje automático.

Figura. 4 Técnicas de Aprendizaje Automático

2.3.3 Técnicas de Análisis Predictivo

De acuerdo a (García & Molina, 2012) se puede definir a la predicción como: “el proceso que intenta determinar los valores de una o varias variables, a partir de un conjunto de datos”. El análisis predictivo es esencialmente un proceso estadístico en que los resultados obtenidos no son precisos, sino que se expresan en términos de probabilidad; esto implica que los niveles de precisión de las predicciones estarán expresados en límites de confianza (Pérez C. , 2004).

En base a la clasificación de técnicas de aprendizaje automático, las técnicas predictivos forman son un subconjunto de las técnicas de aprendizaje supervisado; entre las principales se encuentran: regresión, árboles de predicción y redes de neuronas (Pérez M. , 2014). A continuación se explica brevemente cada las técnicas de Regresión y Árboles de Predicción. La técnica de Redes Neuronales Artificiales se la explica con más profundidad en el siguiente apartado ya que es la que va a ser utilizada en el desarrollo de la investigación.

Regresión

De acuerdo a (Witten, Frank, & Hall, 2011) “en los casos en que se desea predecir un valor numérico y todos los atributos son numéricos la técnica de regresión lineal es una candidata natural a ser considerada en primera instancia”. Cuando los datos no muestran una dependencia lineal, como cuando la variable de respuesta depende de los atributos según una función polinómica se debe considerar el uso de la técnica de regresión no lineal; en este caso se debe realizar ciertas transformaciones a las variables para convertir el modelo no lineal en uno lineal que puede resolverse por el método de mínimos cuadrados (García & Molina, 2012).

Citando a: (García & Molina, 2012) “Un modelo lineal generalizado es el fundamento teórico en la regresión lineal puede aplicarse para modelar las categorías de las variables dependientes”; en estos modelos la variable dependiente no está representada por una variación constante.

Árboles de Predicción

Los árboles de predicción numérica son similares a los árboles de decisión (algoritmos de clasificación); la diferencia radica en que la variable a predecir es continua. En este caso, cada nodo hoja almacena un valor de clase consistente en la media de las instancias que se clasifican con esa hoja, en cuyo caso estamos hablando de un árbol de regresión, o bien un modelo lineal que predice el valor de la clase, y se habla de árbol de modelos. Mientras que en el caso de los árboles de decisión se emplea la entropía de clases para definir el atributo con el que dividir, en el caso de la predicción numérica se emplea la varianza del error en cada hoja (García & Molina, 2012).

Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) son, “redes computacionales que intentan simular, en forma bruta, las redes neuronales del sistema central nervioso biológico (humano o animal)” (Graupe, 2007).

Las RNA son algoritmos computacionales que aprenden de un conjunto de datos y tienen la capacidad de resolver nuevos problemas. El aprendizaje se da como resultado de un proceso de entrenamiento generado a través de un subconjunto de la globalidad de los datos a ser analizados. Las redes neuronales artificiales tienen la capacidad de predecir, clasificar y segmentar datos. En la mayoría de casos pueden tener un rendimiento superior o al menos similar a métodos estadísticos como la regresión (García & Molina, 2012).

Los principios de las RNA fueron formulados por primera vez por McCulloch y Pitts en 1943, en estos principios se asume que las neuronas son elementos binarios; posteriormente (Donald Hebb, 1949) publicó la “Ley de Aprendizaje Hebbian”. En la actualidad el estado del arte del desarrollo de redes neuronales ha avanzado mucho por lo que ya muchos de estos principios no son aplicables.

Las RNA pueden utilizar cualquiera de las formas de aprendizaje es decir, supervisado o no supervisado; esto depende del procedimiento de aprendizaje

utilizado. La dificultad del uso de estos algoritmos se da en la dificultad de acceder y comprender los modelos que generan. Lo que hace especiales a las RNA es el uso de una capa oculta de funciones ponderadas llamadas neuronas, con las que se crea una red que traza otras funciones. Sin la capa oculta las RNA solo serían un conjunto simple de funciones ponderadas (Kirk, 2015). La figura 6, muestra la estructura y componentes de una Red de Neuronas Artificiales.

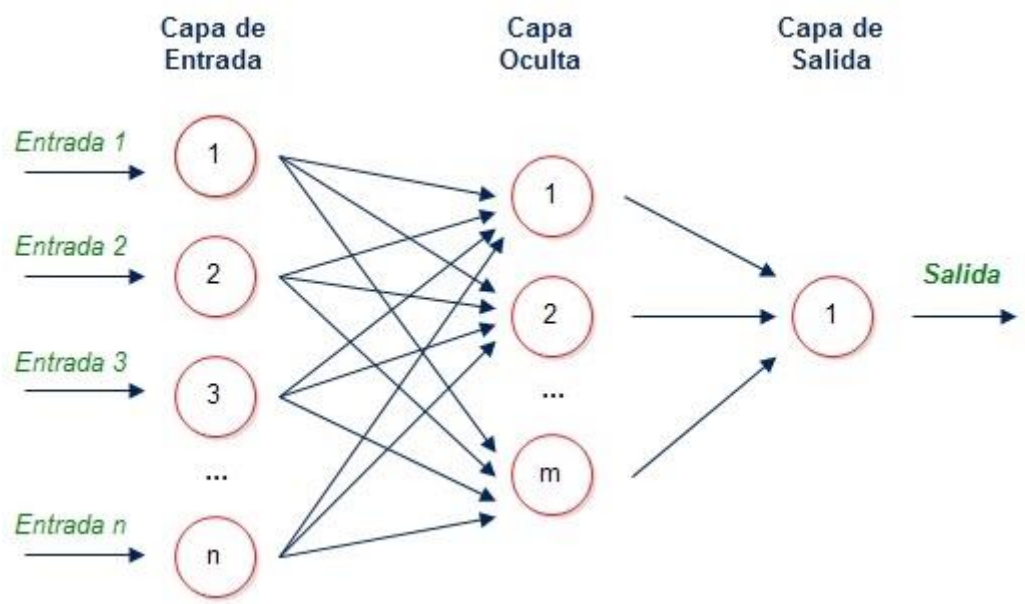

Figura 5 - Estructura de una Red de Neuronas Artificiales

Partes de las Redes Neuronales Artificiales

Capa de Entrada

La capa de entrada no está constituida por neuronas; ya que su propósito principal es el de alimentar a la capa oculta. Se debe tener en cuenta que el tipo de entrada puede ser solo de dos tipos: simétrico o estándar. Las entradas estándar solo pueden tomar dos valores 0 y 1 (falso o verdadero). Las entradas simétricas toma valores entre -1 y 1; en este caso -1 sería falso y 1 verdadero, dando así mejores opciones.

Neuronas

Las neuronas son combinaciones lineales ponderadas que están envueltas en una función de activación. La combinación lineal ponderada (o suma) es una forma de

agregar todos los datos de las neuronas anteriores en una salida para que la siguiente capa consuma como entrada. Las funciones de activación, sirven como una forma de normalizar los datos de modo que sean simétricos o estándar (Kirk, 2015).

Funciones de Activación

Las funciones de activación sirven como una forma de almacenar en memoria intermedia los valores entrantes de cada capa; de esta forma las RNA encuentran patrones dejando de lado el ruido. Hay dos categorías principales de funciones de activación: inclinada o periódica; de forma predeterminada se utiliza las funciones de activación inclinadas. La función Sigmoide es la que se utiliza por defecto debido a su capacidad para suavizar las decisiones (Kirk, 2015).

Capa Oculta

Sin capas ocultas, las Redes Neuronales serían un conjunto de combinaciones lineales ponderadas. En otras palabras, las Redes Neuronales tienen la capacidad de modelar datos no lineales porque hay capas ocultas, cada capa oculta contiene un conjunto de neuronas cuyos resultados alimentan las neuronas de la capa de salida (Graupe, 2007). Cada neurona está conectada a todas las neuronas de las capas anterior y posterior a través de los pesos (García & Molina, 2012).

Capa de Salida

La capa de salida a diferencia de la capa de entrada tiene neuronas; también con valores simétricos o estándar. La capa de salida determina el número de neuronas de salida, el cual depende de lo que se está modelando. La figura YY muestra la composición de una red de neuronas artificiales (Graupe, 2007).

Algoritmos de Entrenamiento

Los algoritmos de entrenamiento se utilizan para definir los pesos de cada neurona; existen algunos algoritmos de entrenamiento y deben ser utilizados de acuerdo al modelo y uso que se desea dar a la red neuronal que se está construyendo; los más conocidos son: Retro-propagación (Back Propagation), QuickProp y RProp; los algoritmos encuentran los pesos óptimos para cada neurona a través de iteraciones. En cada iteración el algoritmo de entrenamiento pasa a través de toda la red y compara el posible valor resultado con el esperado; aprendiendo así de los cálculos erróneos (Kirk, 2015).

Retro-propagación (Back Propagation)

Este algoritmo de entrenamiento "varía los pesos de acuerdo a las diferencias encontradas entre la salida obtenida y la que debería obtenerse; si las diferencias son grandes se modifica el modelo de forma importante y según van siendo menores, se va convergiendo a un modelo final" (García & Molina, 2012). Las iteraciones se detienen cuando el error se mantiene constante de una iteración a otra, obteniendo así un error mínimo. La figura 7, muestra un esquema de este algoritmo.

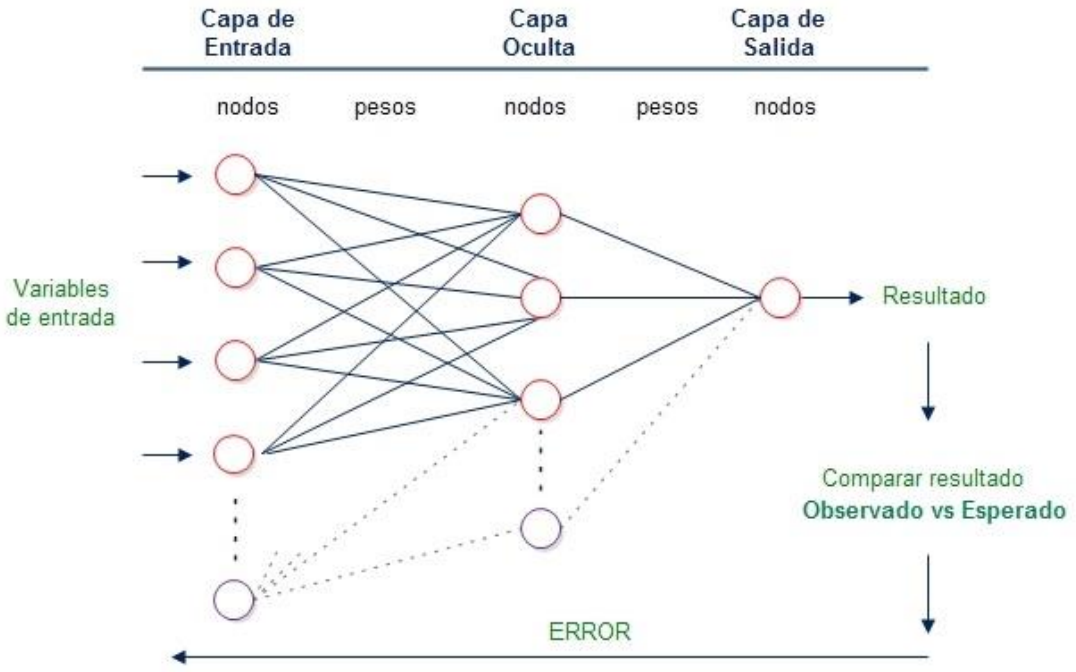

Figura 6 Algoritmo de Retro-Propagación

2.4 Fundamentación Científica de la Variable Dependiente

2.4.1 Gestión de la Ingeniería de Software

La gestión de la ingeniería de software puede definirse como la aplicación de actividades de gestión (planificación, coordinación, medición, supervisión, control y la presentación de informes) para garantizar que los productos de software y los servicios de ingeniería de software se entreguen con eficiencia, eficacia y en beneficio de las partes interesadas (Bourque & Fairley, 2014).

La IEEE Computer Society, define siete temas principales en los que se compone la Administración de la Ingeniería de Software, los dos primeros temas: “Inicialización y Definición de Alcance” y “Planificación del Proyecto de Software” son detallados a continuación.

Inicialización y Definición de Alcance

Está compuesta por un conjunto de actividades cuyo principal objetivo es la determinación de los requisitos del software utilizando para esto varios métodos; además se evalúa la viabilidad del proyecto desde varios puntos de vista. Una vez que se ha establecido la viabilidad del proyecto, las tareas restantes están relacionadas con la especificación de los requisitos y la selección de procesos para la revisión de los mismos (Bourque & Fairley, 2014).

Planificación de Proyectos de Software

Está compuesta por las siguientes actividades: Planificación del Proyecto, Determinación de Entregables, Esfuerzo, Calendarización y Estimación de Costos, Asignación de Recursos, Gestión de Riesgos, Gestión de la Calidad y Gestión del Plan (Bourque & Fairley, 2014).

La secuencia de las actividades mencionadas en el párrafo anterior depende principalmente del modelo de ciclo de vida de desarrollo de software (MCVDS) el

mismo que incluye los procesos de software utilizados para especificar y transformar los requisitos de software en un producto de software entregable (Bourque & Fairley, 2014).

Los MCVDS pueden dividirse en dos categorías: predictivos y adaptativos. Los MCVDS predictivos se caracterizan por el desarrollo de requisitos detallados, planificación detallada y mínima planificación para la iteración entre las diferentes fases. Los MCVDS adaptativos están diseñados para adaptar requerimientos emergentes (aquellos que surgen en la mitad del desarrollo) y realizar un ajuste iterativo de la planificación (Bourque & Fairley, 2014).

En base a la definición anterior se puede deducir que la principal decisión al iniciar un proyecto de software es la de elegir el modelo de ciclo de vida de software que mejor se ajuste al proyecto que se va a realizar teniendo en cuenta sus características iniciales.

2.4.2 Gestión de Proyectos de Software

Para definir la Gestión de Proyectos es necesario en primer lugar conocer la definición de lo que es un proyecto; de acuerdo al Instituto para la Dirección de Proyectos conocido por sus siglas en inglés como PMI: “Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único” (Project Management Institute, 2004).

De esta definición se deduce: que un proyecto tiene principio y fin, dada la naturaleza temporal del mismo; sin embargo, se debe tener en cuenta que la temporalidad no está relacionada a una corta duración. Adicionalmente la definición también nos dice que el fin de un proyecto es el de obtener un producto, servicio o resultado único.

De acuerdo al PMI la Gestión o Dirección de Proyectos se define como: “la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades

del proyecto para cumplir con los requisitos del mismo" (Project Management Institute, 2004).

Si bien el PMI da principios generales relacionados con las buenas prácticas en la Gestión de Proyectos, no hace énfasis especial en los proyectos de software. De acuerdo a la IEEE Computer Society en la ingeniería de software se distinguen diferentes tipos de proyectos como por ejemplo: desarrollo de producto, servicios de consultoría, mantenimiento de software, creación de servicios, implantación de productos, entre otros. Durante el ciclo de vida de un producto de software este puede requerir de la creación de varios proyectos relacionados (Bourque & Fairley, 2014).

Según el PMI todos los proyectos pueden ser gestionados de forma similar, esto también incluye a los proyectos de software; sin embargo, se debe tener en cuenta que existen ciertos aspectos muy específicos de los proyectos de software y de su ciclo de vida, que pueden complicar la gestión. De acuerdo a (Bourque & Fairley, 2014), se debe tener en consideración los siguientes aspectos al momento de gestionar proyectos de software:

- Los clientes a menudo no saben lo que se necesita o lo que es factible realizar.
- A menudo, los clientes no aprecian las complejidades inherentes a la ingeniería de software, particularmente en lo que respecta al impacto en los cambios de los requisitos.
- Es probable que una mayor comprensión y condiciones cambiantes generen nuevos requisitos o cambios profundos en los mismos.
- Como resultado de los requisitos cambiantes, el software suele construirse utilizando un proceso iterativo en lugar de una secuencia de tareas cerradas.
- La ingeniería de software incorpora necesariamente creatividad y disciplina. Mantener un equilibrio adecuado entre los dos es a veces difícil.
- El grado de novedad y complejidad suele ser elevado.
- A menudo hay una rápida tasa de cambio en la tecnología subyacente.

La planificación de proyectos es uno de los trabajos más importantes de un gerente de proyectos de software. Esta actividad consiste en dividir el trabajo en actividades y

asignarlas a los miembros del equipo del proyecto, anticipar los problemas que puedan surgir y preparar soluciones provisionales a esos problemas (Sommerville, 2011).

De acuerdo a (Sommerville, 2011) “la planificación del proyecto toma lugar en tres etapas del ciclo de vida de un proyecto de software” las cuales se describen a continuación:

1. En la elaboración de la propuesta, cuando se está realizando una oferta para un contrato de desarrollo, prestación de servicios o implantación de un producto de software. La planificación es útil para verificar los recursos necesarios, estimar el tiempo y el precio que se debe cotizar al cliente.
2. Durante la planificación inicial del proyecto, cuando se tiene que planificar quién trabajará en el proyecto, como se dividirá la totalidad del proyecto en incrementos y como se asignarán los recursos. En esta etapa se tiene más información que en la anterior por lo que se puede realizar afinaciones y cambios a la estimación inicial de esfuerzo que se realizó en la fase de elaboración de la propuesta.
3. Periódicamente durante todo el proyecto, con el avance del trabajo realizado el líder del proyecto tiene más conocimiento del sistema que está implementando y de las capacidades de su equipo de desarrollo. Esta información permite realizar estimaciones más precisas de cuánto durará el trabajo las cuales deben ser trasladadas al cronograma del proyecto para levantar alertas tempranas sobre posibles retrasos.

Para una correcta y eficaz gestión de un proyecto de software el líder de proyecto debe centrarse en cuatro componentes principales que son: personas, productos procesos y personas; teniendo en cuenta el orden de los mismos (Pressman, 2010).

La necesidad de contar con un equipo motivado y altamente calificado es algo que se ha venido discutiendo desde los años 60 (Jones C., 2007). “Cada organización necesita mejorar continuamente su capacidad para atraer, desarrollar, motivar, organizar y retener la fuerza de trabajo necesaria para lograr sus objetivos estratégicos de negocio” (Curtis, Hefley, & Miller, 2001).

Antes de planificar un proyecto, deben establecerse objetivos y alcances del producto, considerar soluciones alternativas e identificarse las limitaciones técnicas y de gestión; sin esta información, es imposible definir estimaciones razonables (y exactas) del costo (Pressman, 2010).

El proceso de software proporciona el marco desde el cual se puede establecer un plan integral para el desarrollo de software. Un pequeño número de actividades marco son aplicables a todos los proyectos de software, independientemente de su tamaño o complejidad (Pressman, 2010).

Un proyecto planificado y controlado es la única forma en la que se puede gestionar la complejidad. Para evitar el fracaso de un proyecto el líder y los ingenieros de software encargados de la construcción del producto deben comprender los factores críticos de éxito que conducen a una buena gestión de proyectos y desarrollar un enfoque de sentido común para la planificación (Pressman, 2010).

2.4.3 Estimación del Esfuerzo en Proyectos de Software

La estimación del esfuerzo para un proyecto es uno de los procesos de la “Gestión del Tiempo” definido por el PMI (Project Management Institute, 2004). La entrada principal de este proceso es la lista de definición de las actividades que conforman el proyecto y su respectiva secuencia. Una vez que se tiene la estimación del esfuerzo para cada actividad se puede asignar los recursos que serán los encargados de llevarlas a cabo. En este caso las actividades son las acciones específicas que se deben llevar a cabo para generar los entregables del proyecto (Project Management Institute, 2004).

El proceso de estimación del esfuerzo es una de las principales preocupaciones de los líderes de proyectos, de esta tarea depende en gran parte el costo del proyecto y la fecha de finalización del mismo. Citando a (Pressman, 2010) “La estimación de recursos, costos y programación para un esfuerzo de ingeniería de software requiere experiencia, acceso a buena información histórica (métricas) y el coraje de

comprometerse a predicciones cuantitativas cuando la información cualitativa es todo lo que existe.”

Existen dos factores en los proyectos que pueden influir de manera directa en la estimación del esfuerzo de un proyecto, estas son: la complejidad y el tamaño del proyecto. La complejidad tiene un fuerte efecto sobre la incertidumbre inherente a la planificación; sin embargo, es una medida relativa que se ve afectada por la familiaridad con el esfuerzo pasado. El tamaño es otro factor importante que puede afectar la precisión y la eficacia de las estimaciones. A medida que el tamaño aumenta, la interdependencia entre elementos crece rápidamente lo que complica la descomposición del problema (Pressman, 2010).

Los proyectos de desarrollo de software están definidos por una especificación de requisitos. Los requisitos para un sistema de software son las descripciones de lo que el sistema debe hacer, los servicios que proporciona y las limitaciones en su funcionamiento. Estos requisitos reflejan las necesidades de los clientes para un sistema que cumple un determinado propósito; es por esto que la estimación del esfuerzo se la realiza en base al trabajo necesario para el cumplimiento de cada uno de los requisitos (Sommerville, 2011).

La estimación de costos y esfuerzo en el software nunca será una ciencia exacta. Existen demasiadas variables (humanas, técnicas, ambientales, políticas) pueden afectar el costo final del software y el esfuerzo aplicado para desarrollarlo. Sin embargo, para lograr una mejor precisión en la estimación se puede seguir una serie de pasos sistemáticos que proporcionan estimaciones con un riesgo aceptable (Pressman, 2010).

A continuación se explican cuatro diferentes opciones que propone (Pressman, 2010) que pueden ayudar a obtener estimaciones confiables de costo y esfuerzo:

1. Atrasar la estimación hasta tener todos los requisitos detallados y revisados. Esta opción lastimosamente no es viable, las estimaciones de costo y tiempo deben ser proporcionadas por adelantado.

2. Utilizar información de estimaciones de proyectos similares que ya se han completado. Esta opción tiene grandes posibilidades de éxito cuando el proyecto a ser estimado es bastante parecido a proyectos realizados anteriormente y no se vea afectado por factores externos como cliente, condiciones comerciales, legislación y cultura propia de cada país.
3. Utilizar técnicas de descomposición relativamente simples para generar estimaciones de costo y esfuerzo del proyecto. Para esta opción también es necesario contar con los requisitos del proyecto definidos con un alto nivel de detalle
4. Utilizar uno o más modelos empíricos para la estimación del costo y esfuerzo del software. Esta técnica es utilizada por grandes empresas de software; adicionalmente de que existe mucha información al respecto; sin embargo, no es utilizada en empresas pequeñas o medianas ya que no cuentan con profesionales especializados para aplicar los modelos.

En los proyectos de software por lo general los requisitos no son divisibles a diferencia de ciertas actividades de otros proyectos, la división de actividades es una técnica muy común utilizada en la gestión de proyectos; sin embargo, su uso en los proyectos de software depende de la experiencia del líder del proyecto (Bourque & Fairley, 2014).

La secuencia de las actividades es el proceso que consiste en identificar y documentar las relaciones entre las actividades del proyecto. El beneficio clave de este proceso reside en la definición de la secuencia lógica de trabajo para obtener la máxima eficiencia teniendo en cuenta todas las restricciones del proyecto (Project Management Institute, 2004). En la gestión de proyectos de software; también debe realizarse esta actividad teniendo en cuenta el orden en que debe ser desarrollado cada requisito funcional; evitando así posibles pérdidas de tiempo en el equipo de desarrollo (Sommerville, 2011).

De acuerdo al (Project Management Institute, 2004) “La estimación de la duración de las actividades es el proceso de realizar una estimación de la cantidad de períodos de trabajo necesarios para finalizar las actividades individuales con los recursos

estimados.” La estimación de la duración depende de los recursos disponibles para completar cierta actividad; en los proyectos de desarrollo de software el principal recurso son las personas (ingenieros de desarrollo); sin embargo, y a diferencia de otros tipos de proyectos, primero se estima la duración del esfuerzo de desarrollo para luego asignar las personas que se encargaran de realizar el trabajo necesario para cumplir con un requisito, adicionalmente se debe tener en cuenta que la adición de personas en el desarrollo no siempre producirá una disminución directamente proporcional en la duración de la actividad (Pressman, 2010).

Para los proyectos de desarrollo de software la estimación del esfuerzo de cada requisito generalmente se la realiza en “horas-hombre”, es decir se estima el tiempo que le tomará a un ingeniero de desarrollo el cumplir con el trabajo necesario para finalizar la implementación de un requisito. En base a las características propias del requisito y los recursos disponibles se define la duración del desarrollo del requisito. La asignación de recursos (personas) para el desarrollo de una actividad depende de dos factores: la experiencia profesional (consultor, arquitecto, programador senior, programador junior, pasante, etc.) y de la especialización (arquitecto, administrador de base de datos, analista funcional, desarrollador, diseñador de interfaces, documentador, etc.) (Pressman, 2010).

Se debe tener en cuenta que los requisitos y diseño de proyectos de software tienden a cambiar de forma significativa durante el ciclo de desarrollo; de hecho la tasa promedio de nuevos requisitos después de la fase inicial crece en promedio 2% por mes calendario; algunos proyectos han observado un crecimiento de requisitos de más de 5% por mes durante las fases posteriores de diseño y codificación (Jones C., 2007).

Como se ha señalado los proyectos de software, tienen ciertas características que los hacen diferentes es por esto que la Ingeniería de Software se ha encargado de realizar diversos estudios encaminados a determinar la mejor forma de estimación del esfuerzo para completar una tarea, en la actualidad se dispone de varias técnicas de estimación que pueden ser utilizadas por los líderes de proyectos (Jones C., 2007).

Si bien las estimaciones de costos y fechas límites de proyectos de software deberían ser precisas en un mundo ideal; sin embargo, revisando los resultados reales, es recomendable ser más conservadores que optimistas (Jones C., 2007). Una de las quejas principales acerca de los proyectos de software es su tendencia angustiosa a exceder costos y fechas límite planeados. Lastimosamente clientes y directivos tienden a ejercer presión considerable en gerentes y empleados a cargo de la realización de estimaciones apuntando hacia evaluaciones optimistas. Es por esto que toda estimación debe ser capaz de defenderse sola. La mejor defensa es una buena colección de datos históricos basada en proyectos similares (Jones C., 2007).

En base a los conceptos analizados se puede afirmar que el estimado de esfuerzo requerido para un proyecto o partes de un proyecto puede determinarse usando un modelo de estimación ajustado y basado en datos históricos de tamaño y esfuerzo (cuando estén disponibles) y otros métodos relevantes tales como juicio de expertos y analogía. Las dependencias de tareas se pueden establecer y las oportunidades potenciales para completar las tareas concurrentemente y secuencialmente se pueden identificar y documentar usando un diagrama de Gantt (Bourque & Fairley, 2014).

Técnicas y Modelos de Estimación

Como se menciona en la sección anterior, la estimación del cronograma del proyecto es difícil. Lo más probable es que se tenga que hacer estimaciones iniciales sobre la base de una definición de requisitos de usuario de alto nivel. Existe un grado de incertidumbre respecto a las personas que serán asignadas al proyecto y posiblemente no se conozcan sus habilidades técnicas. Hay tantas incertidumbres que es imposible estimar con precisión los costos de desarrollo del sistema durante las primeras etapas de un proyecto (Sommerville, 2011).

Estimación del tamaño del producto

La mayoría de los modelos, herramientas y técnicas de estimación utilizan alguna medida de tamaño como atributo fundamental para basar las estimaciones.

Dependiendo de la naturaleza del producto, los factores adicionales al tamaño como: complejidad, interfaces externas, cantidad de datos a manipular, entre otras; pueden influir en las estimaciones de esfuerzo, calendario, recursos, factores de calidad y costo. Estos factores adicionales se incluyen como “factores de ajuste” en la mayoría de los modelos de estimación (Fairley, 2009).

El “tamaño” es el atributo principal de los métodos de estimación, porque el mismo tiene un mayor grado de relación con el esfuerzo y costo; además de que su medición tiene un mayor grado de objetividad respecto a otros atributos y los datos relacionados a tamaño y esfuerzo son fácilmente recopilados y pueden ser almacenados para su uso posterior (Fairley, 2009).

Históricamente, las líneas de código fuente (LOC) se han utilizado como medida de tamaño (Jones C., 2007); sin embargo, en la actualidad las empresas han dejado de utilizar esta medida por las razones que son claramente explicadas por (Fairley, 2009) y que se describen a continuación:

- Es difícil estimar las líneas de código en fases tempranas de un proyecto; adicionalmente es difícil relacionar los cambios a los requerimientos con los cambios en las líneas de código estimadas;
- Calcular la productividad como líneas de código generadas por programador/mes no favorece a la reutilización y calidad del software
- Los métodos modernos de desarrollo, como el desarrollo impulsado por modelos o prototipos, los lenguajes de programación de paradigmas funcionales o de objetos, la reutilización de componentes y librerías propias o de código abierto, hacen que la relación entre líneas de código y atributos del proyecto sea menos relevante y menos precisa que en el pasado.

La medida de tamaño basada en puntos de función es la más utilizada en la actualidad. Los puntos de función (FP) se calculan contando el número de diferentes tipos de entradas, salidas, archivos internos, consultas e interfaces en un sistema que se va a estimar (Pressman, 2010). Estos recuentos se basan en reglas de conteo

objetivo y cada entrada, salida, archivo interno, consulta e interfaz únicos se ponderan como simple, media o compleja. Los valores ponderados se suman para proporcionar un número total de puntos de función no ajustados; finalmente se aplican factores de ajuste como la complejidad del procesamiento, la tasa de transacción y la facilidad de uso requerida para tener en cuenta las condiciones que requerirán más o menos esfuerzo que el proyecto típico (Fairley, 2009).

Clasificación de las técnicas de Estimación

El PMI menciona algunas técnicas que los gerentes de proyecto pueden utilizar para la estimación del esfuerzo en las actividades de sus proyectos, entre las más conocidas y utilizadas se tiene: Juicio de Expertos, Estimación Análoga, Estimación Paramétrica y Estimación por Tres Valores. (Project Management Institute, 2004), a continuación se describe cada una de estas técnicas:

Juicio de Expertos.- Se guía en la información histórica y puede proporcionar información sobre la estimación de la duración recomendada en base a proyectos anteriores.

Estimación Análoga.- Es una técnica para estimar la duración o el costo de una actividad o de un proyecto mediante la utilización de datos históricos de una actividad o proyecto similar. La estimación análoga utiliza parámetros de un proyecto anterior similar, tales como duración, presupuesto, tamaño, carga y complejidad, como base para estimar los mismos parámetros o medidas para un proyecto futuro. La estimación análoga es menos costosa y requiere menos tiempo que otras técnicas, pero también es menos exacta.

Estimación Paramétrica.- Es una técnica de estimación en la que se utiliza un algoritmo para calcular el costo o la duración sobre la base de los datos históricos y los parámetros del proyecto. La estimación paramétrica utiliza una relación estadística entre datos históricos y otras variables propias del proyecto para calcular una estimación de los parámetros de una actividad tales como costo, presupuesto y duración. Con esta técnica pueden lograrse niveles superiores de exactitud, dependiendo de la sofisticación y de los datos que utilice el modelo.

Estimación por Tres Valores.- Esta estimación utiliza tres estimaciones para definir un rango aproximado de duración de una actividad:

- Más probable (tM). Esta estimación se basa en la duración de la actividad, en función de los recursos que probablemente sean asignados, de su productividad, de las expectativas realistas de disponibilidad para la actividad, de las dependencias de otros participantes y de las interrupciones.
- Optimista (tO). Estima la duración de la actividad sobre la base del análisis del mejor escenario posible.
- Pesimista (tP). Estima la duración de la actividad sobre la base del análisis del peor escenario posible.

En base a estos tres valores se calcula la duración de la actividad utilizando la siguiente formula:
tE = (tO + 4tM + tP) / 6

En proyectos de software se utiliza técnicas similares a las definidas por el PMI para la generalidad de los proyectos; sin embargo, la Ingeniería de Software se ha encargado de especializarlas para su uso en proyectos de software y adicionalmente ha definido otras que se explican a continuación.

La clasificación de las técnicas de estimación para proyectos de software tiene ligeras variaciones de acuerdo a diferentes autores, en este trabajo presentamos dos clasificaciones diferentes la primera de acuerdo a Ian Sommerville y la segunda de acuerdo a Richard Fairley.

Para realizar la estimación se pueden utilizar 2 tipos de técnicas de acuerdo a (Sommerville, 2011):

1. Técnicas basadas en la experiencia: La estimación del esfuerzo de los requisitos se basa en la experiencia del gerente en proyectos anteriores y en el dominio de la aplicación.

2. Modelado Algorítmico de Costos: En este enfoque, se utiliza un enfoque de fórmula para calcular el esfuerzo del proyecto basado en estimaciones de los atributos del producto, como el tamaño y las características del proceso, como la experiencia del personal involucrado.

En ambos casos, el gerente del proyecto debe usar su juicio para estimar el esfuerzo directamente o estimar las características del proyecto y del producto. En la fase inicial de un proyecto, estas estimaciones tienen un amplio margen de error; sin embargo, estas se hacen cada vez más precisas a medida que avanza el Proyecto (Sommerville, 2011).

De acuerdo a (Fairley, 2009) “Las diversas técnicas para estimar los atributos de interés para proyectos de software (por ejemplo, esfuerzo, calendario, recursos, factores de calidad y costo) pueden clasificarse como pragmáticas, basadas en la teoría y basadas en regresión.”

Las técnicas de estimación pragmática incluyen: Regla de Oro, Analogía, Juicio Experto, Delphi, WBS / CPM / PERT; las técnicas de estimación basadas en la teoría incluyen: Sistemas Dinámicos. SLIM y finalmente los modelos de estimación basados en regresión incluyen: COCOMO, Modelos Derivados Localmente. Los modelos basados en la teoría y en la regresión usan el tamaño como la entrada principal de la estimación; las técnicas pragmáticas pueden o no utilizar una medida de tamaño (Fairley, 2009).

De las dos clasificaciones anteriores se ha seleccionado la de Richard Fairley como guía para la fundamentación teórica del trabajo de investigación; a continuación se procede a describir cada una de las diferentes técnicas; el conocimiento de la mismas ayudará a tener una mejor idea de cuáles son las técnicas que se puede utilizar en etapas tempranas de la planificación de proyectos de software teniendo en cuenta las características de la industria local. La ***Figura 7***, muestra una clasificación más detallada de las técnicas de estimación.

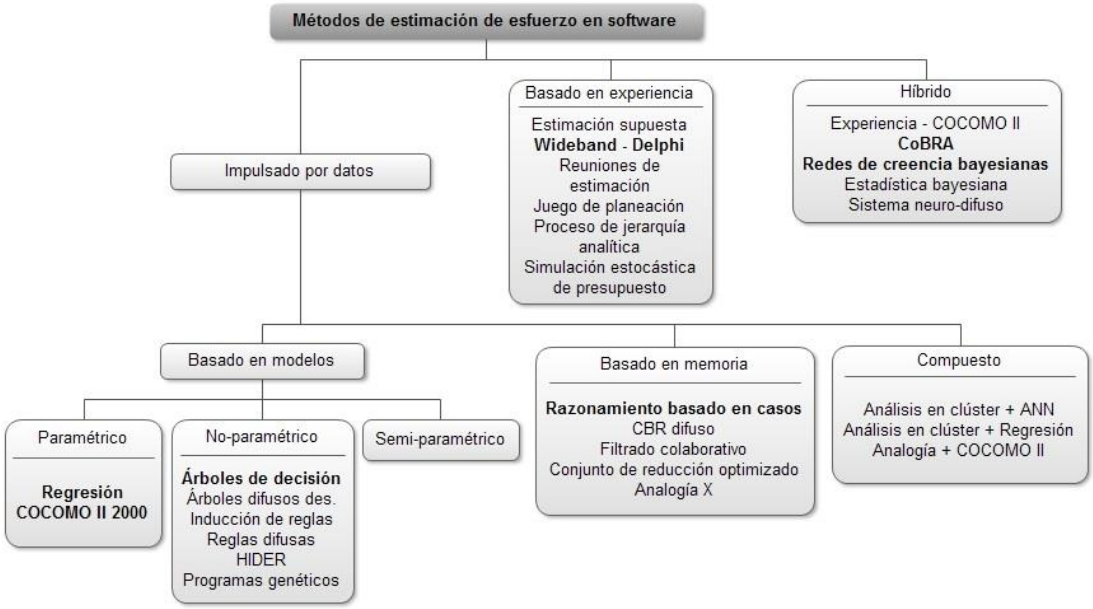

Figura 7 - Clasificación de los Métodos de Estimación de Esfuerzo

Técnicas de Estimación Pragmáticas

Estas reglas se caracterizan por no estar basadas en modelos teóricos o análisis de regresión. Aunque puede creerse que al no estar basadas en métodos estadísticos estas técnicas no son confiables; la correcta utilización de las mismas puede llegar a ser de gran utilidad en algunos proyectos de software.

Regla de Oro

La regla de oro conocida por sus siglas como (ROT) es una directriz generalmente aceptada por la industria; las reglas de oro provienen de años de experiencia y práctica de la industria y que no poseen un fundamento o base formal. Estas reglas pueden ser aplicables a diferentes factores del proyecto como la productividad, la complejidad o el aseguramiento de la calidad. Si bien existen muchas reglas de oro utilizadas internacionalmente es recomendable que cada empresa ajuste dichas reglas a su propia realidad (Fairley, 2009).

La estimación basada en el uso de reglas de oro, no basta para estimar todo el esfuerzo de un proyecto de software; sin embargo, su utilización puede ayudar a incrementar ciertos tiempos que podrían no ser considerados inicialmente, un ejemplo

podría ser: "*El tiempo necesario para el proceso de pruebas y correcciones es del 20% adicional al tiempo total del esfuerzo estimado en para el desarrollo de una funcionalidad*".

Analogía

La analogía es una técnica ampliamente utilizada por diferentes empresas que permite la estimación de valores para ciertos atributos de diferentes tipos de proyectos; esta técnica no es exclusiva de la industria del software. El objetivo de esta técnica es el de "encontrar" productos similares (análogos) que se hayan desarrollado en su empresa o comunidad local y de los cuales se tenga la información disponible para los atributos que se desea medir. El éxito de esta estimación radica en lo cercana que es la analogía al proyecto actual (Fairley, 2009).

Las estimaciones basadas en analogías, no son utilizadas para estimar todo el esfuerzo necesario para un proyecto; sin embargo, pueden ayudar con la estimación de ciertas partes del mismo. Para sacar el máximo provecho a esta técnica es necesario que las empresas lleven registros detallados de sus proyectos en repositorios de datos, los cuales puedan ser consultados fácilmente (Bourque & Fairley, 2014).

Delphi

La técnica Delphi se utiliza para obtener estimaciones compuestas de diferentes expertos en forma anónima. El proceso puede resumirse así; el líder de proyecto pide a cada experto por separado sus estimaciones para lo cual debe proveer de toda la información referente al proyecto con la que cuenta. Cada experto tiene un tiempo prudencial para realizar sus estimaciones con la justificación respectiva en caso de ser necesario y presentarlas al líder del proyecto. El líder de proyecto se encarga de consolidar los resultados; para casos en los que se encuentren divergencias en las estimaciones el líder deberá organizar otra ronda de estimación; proveyendo esta vez los resultados anteriores de cada experto; por lo general los resultados deberían

converger luego de 3 o 4 rondas; en caso de no hacerlo el líder de proyecto debe convocar a una reunión con los expertos para llegar a consensos (Scott, 2001).

Entre las ventajas de utilizar este método podemos mencionar el de obtener diferente opiniones sin influencias gracias a la anonimidad en la presentación de los resultados; adicionalmente el uso de más de una ronda de estimación permite flexibilizar las opiniones de cada experto. Por otro lado las debilidades de este método se centran en el tiempo y dependencia de cada experto para la entrega de sus estimaciones y la probable inflexibilidad o influencia indebida en las reuniones de consenso por parte de algunos expertos (Fairley, 2009).

WBS/CPM/PERT

Esté método se basa en la descomposición del alcance de un proyecto en paquetes de trabajo (actividades y tareas), la estimación del esfuerzo se la realiza en los niveles interiores, para esto se puede utilizar cualquiera de la técnicas detalladas anteriormente. La estimación final del esfuerzo está dada, por la suma de los esfuerzos de cada paquete de trabajo. Esta técnica es posiblemente la que otorga un mayor grado de precisión ya que la estimación se la realiza en base al detalle y descomposición de los requisitos del proyecto (Fairley, 2009).

La principal ventaja del uso de esta técnica es la mayor precisión de las estimaciones, acompañado de un mayor detalle en la descomposición del alcance del proyecto. Se debe tener en cuenta que esta técnica no puede ser aplicada si no se conoce a suficiente nivel de detalle el alcance del proyecto; así como las relaciones que deben mantenerse en la secuenciación de las actividades (Scott, 2001).

Modelos de estimación basados en la teoría.

Un modelo de estimación basado en la teoría se llama así porque existe una teoría subyacente de los proyectos de software en los que se basa cada modelo de estimación. En la actualidad se tiene principalmente dos modelos basados en la teoría: Dinámica de Sistemas y SLIM; a continuación se explica brevemente cada uno de estos modelos.

Dinámica de sistemas

La dinámica de sistemas es una metodología para analizar y modelar el comportamiento temporal en entornos complejos. Se basa en la identificación de los bucles de realimentación entre los elementos, y también en las demoras en la información y materiales dentro del sistema (Forrester, 1960).

Los modelos de la dinámica de sistemas se expresan en un conjunto de ecuaciones diferenciales cuyas variables son las propiedades del sistema modelo. El éxito del modelo está en encontrar las variables críticas del sistema complejo e identificar los vínculos causales que existen entre ellas. Para la construcción de estos modelos se necesita de expertos matemáticos y estadísticos, además de expertos en el dominio del proyecto que se está estimando (Izquierdo, Ordax, Santos, & Martínez, 2008).

Modelo SLIM

El Modelo SLIM, abreviación del inglés “Software LIfecycle Management”, o modelo Putnam es una técnica de estimación de costos para proyectos de software. Es una de las técnicas que más repercusión ha tenido en el mundo de la ingeniería del software desde su definición en la década de los 70 (Velthuis, Villalón, Bravo, & Sanz, 2003).

El modelo está basado en dos ecuaciones, con dos variables desconocidas: tiempo y esfuerzo. Lastimosamente uno de los componentes del modelo es el número de líneas de código, medida de tamaño que ha quedado en desuso; las principales fortalezas del modelo es la solución simultánea de esfuerzo y duración y el uso de la simulación de Monte Carlo, lo que le permite ofrecer diferentes combinaciones de esfuerzo y tiempo con diferentes valores de probabilidad (Fairley, 2009).

Modelos de estimación basados en Regresión

Los modelos de estimación basados en regresión se basan en ecuaciones derivadas de datos históricos obtenidos de proyectos anteriores. La construcción de

ecuaciones a partir de los datos se conoce como análisis de regresión el cual es un proceso estadístico. Las ecuaciones pueden incorporar múltiples variables independientes; sin embargo, en la mayoría de los casos estos modelos se basan en el análisis de la relación entre una única variable independiente (por ejemplo, el tamaño) y una variable dependiente (por ejemplo, esfuerzo) (Fairley, 2009).

El uso de tamaño como variable independiente se debe a que la estimación de este atributo puede ser más objetivo, como en el caso de Puntos de Función y Líneas de Código. Otros autores clasifican a estos modelos con el nombre de “Modelos Empíricos”. A continuación se describe los modelos como COCOMO, estimación Monte Carlo y calibración local.

Modelo COCOMO

El modelo COCOMO fue definido y descrito por Barry Boehm en su libro “Economía de la Ingeniería del Software”. El nombre COCOMO es un acrónimo de “COnstrucive COst MOdel” en español (Modelo Constructivo de Costo). El modelo COCOMO incluye tres submodelos de estimación cada uno de los cuales ofrece un mayor nivel de precisión en el cálculo de la estimación del esfuerzo; el autor los define como: Modelo 1 o Básico el cual calcula el esfuerzo en función del tamaño del programa, Modelo 2 o intermedio calcula el esfuerzo en función del tamaño y de un conjunto de conductores de costo que incluyen evaluaciones subjetivas del producto, hardware y personal; Modelo 3 o avanzado este modelo toma las características del modelo 2 y lleva a cabo un evaluación de los conductores de diseño en cada una de las fases del ciclo de vida del producto (Pressman, 2010).

El modelo COCOMO ha sufrido algunas variaciones a lo largo de los últimos años gracias a varias investigaciones académicas; sin embargo y aunque es muy reconocido y utilizado por la industria, la calidad de sus resultados dependen de la correcta estimación del tamaño del producto que se está construyendo y de la correcta valuación de los “conductores de costo”; como ya se mencionó anteriormente el uso de líneas de código o puntos de función para la estimación de tamaño no son

actualmente utilizadas por empresas pequeñas o medianas; lo que imposibilita su utilización (Jones C., 2007).

Simulación Montecarlo

El método Montecarlo es un método numérico que permite resolver problemas físicos y matemáticos mediante la simulación de variables aleatorias, su uso en la estimación de esfuerzo para proyectos se lo realiza utilizando aplicaciones especiales u hojas de cálculo las cuales se encargan de realizar este tipo de simulación obteniendo diferentes distribuciones de probabilidad que son las utilizadas para calcular el valor más probable del esfuerzo. Este método se basa en repetir cientos o miles de veces la simulación de los valores para generar un histograma de esfuerzo probable (Fairley, 2009).

De acuerdo a lo revisado se puede afirmar que existen varias técnicas que los líderes de proyecto tienen a disposición para el cálculo del esfuerzo en sus proyectos. La selección de una técnica de estimación depende varios factores como: el nivel de detalle de los requisitos del software, la composición del equipo del proyecto con sus respectivas habilidades, la complejidad del proyecto, el presupuesto y la finalmente el contar con la experiencia o personal experimentado en el uso de una determinada herramienta de estimación (Jones C., 2007).

El contar con información histórica de los proyectos realizados es un componente clave para el proceso de estimación, es por esto que las empresas de software independiente de su tamaño deberían establecer y asegurar en sus políticas la recolección y registro de todos los datos relevantes de sus proyectos (Jones C., 2007).

Con la necesidad de realizar ofertas en base a requisitos de alto nivel y sin detalle el no tener información histórica imposibilita la tarea, por otro lado, el contar con información histórica y no emplear métodos estadísticos y matemáticos que ayuden en su proceso conducirá a estimaciones con un alto nivel de imprecisión lo cual puede conducir al fracaso del proyecto.

CAPITULO III
METODOLOGÍA

3.1 Introducción

La presente investigación se basa en la exploración de la información histórica de tiempos reales de desarrollo de software de los proyectos de implantación de la empresa Gestor; es por esto que se adoptará una metodología de Minería de Datos. A continuación, se describe la metodología seleccionada como un paso previo a su aplicación en los siguientes apartados de este documento.

Antes de describir la metodología seleccionada y para este caso en particular es necesario establecer la diferencia entre modelo de procesos y metodología; “la diferencia fundamental entre metodología y modelo de proceso radica en que el modelo de proceso establece qué hacer, y la metodología especifica cómo hacerlo” (Moine, Haedo, & Gordillo, 2011). La metodología además de establecer las fases de un proceso define las tareas que se deben ejecutar y como estas deben llevarse a cabo. La explicación anterior es una introducción necesaria, ya que muchas organizaciones y autores no establecen esta diferencia y piensan que la modelo de extracción de conocimiento en bases de datos (KDD) es una metodología y la utilizan como referencia para sus proyectos de minería de datos.

Desde el año 2000, se han propuesto diversas metodologías para el desarrollo de proyectos de minería de datos como SEMMA, Catalyst (conocida como P3TQ) y CRISP-DM (Arancibia, 2009); de acuerdo a un estudio publicado por la comunidad KDnuggets en el año 2007, la metodología más utilizada en los proyectos de minería de datos es CRISP-DM, la cual de acuerdo al estudio fue utilizada en el 42% de las proyectos tomados como base del estudio (Moine, Haedo, & Gordillo, 2011). KDnuggets volvió a realizar una encuesta en el año 2014 y se comprobó que a pesar de que transcurrieron siete años los resultados fueron similares, en una muestra de 200 participantes el 43% de los mismos respondió que utiliza la metodología CRISP-DM (Piatetsky, 2014).

3.2 Metodología CRISP-DM

Esta metodología fue creada en el año 2000 por el grupo de empresas SPSS, NCR y Daimer Chrysler; la metodología no fue creada de forma teórica o académica; sino en base a la experiencia acumulada por anteriores proyectos de minería de datos que habían sido ejecutados por las organizaciones que la definieron (Moro, Laureano, & Cortez, 2011). La metodología comprende seis fases: Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación e Implantación; estas fases están compuestas por varias tareas y sub-tareas; que son las que definen las actividades a ser realizadas. Adicionalmente se debe tener en cuenta que la metodología tiene la característica de ser iterativa; lo que permite si es necesario regresar a fases o tareas ya ejecutadas para mejorar el resultado final (Chapman, y otros, 2000).

La metodología CRISP-DM está conformada por un modelo de procesos jerárquico compuesto por un conjunto de tareas descrito en cuatro niveles de abstracción: fase, tarea genérica, tarea especializada e instancia de proceso tal como se muestra en la ***Figura 8***. Las fases y las tareas genéricas se pretende que cubran la mayoría de proyectos de minería de datos. El tercer nivel define las tareas especializadas que son las que descubren las acciones que deben ejecutarse en situaciones específicas; el cuarto nivel que corresponde a las instancias de proceso es un registro de las acciones, decisiones y resultados del proyecto de minería de datos (Chapman, y otros, 2000).

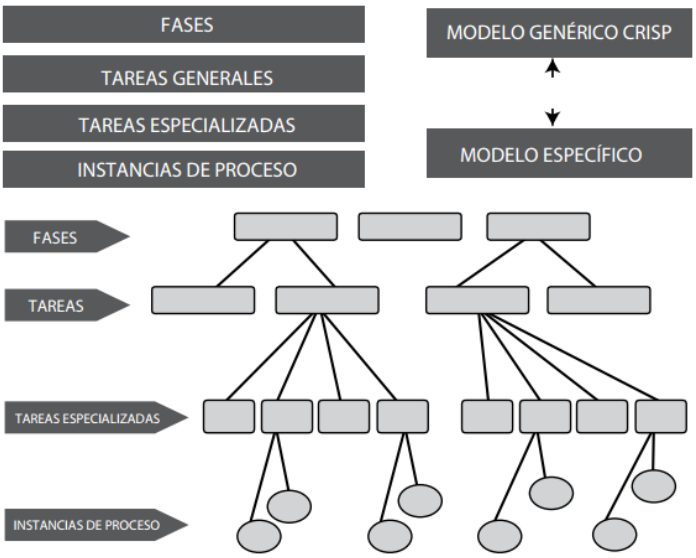

Figura 8 Esquema de los cuatro niveles de CRISP-DM. Tomado de CRISP-DM 1.0 Step-by-step data mining guide (Chapman, y otros, 2000).

"En el nivel superior, el proceso de minería de datos es organizado en un número de fases; cada fase consiste de varias tareas genéricas de segundo nivel. Este segundo nivel lo llaman genérico porque está destinado a ser bastante general para cubrir todas las situaciones posibles de minería de datos. Las tareas genéricas están destinadas a ser tan completas y estables como sea posible" (Chapman, y otros, 2000).

El tercer nivel, llamado de "*Tareas Especializadas*" permite describir como se deben realizar las tareas en situaciones específicas. El cuarto nivel, "*Instancias de Proceso*", es un registro de las acciones, decisiones, y de los resultados del proceso (Chapman, y otros, 2000).

3.2.1 El modelo de Referencia CRISP-DM

El modelo de procesos proporciona una descripción del ciclo de vida del proyecto de minería de datos. Este contiene las fases del proyecto con sus respectivas tareas y las relaciones entre las mismas. El ciclo de vida del proyecto de minería de datos de acuerdo a la metodología CRISP-DM consiste en seis fases, el cual se muestra en la ***Figura. 9***.

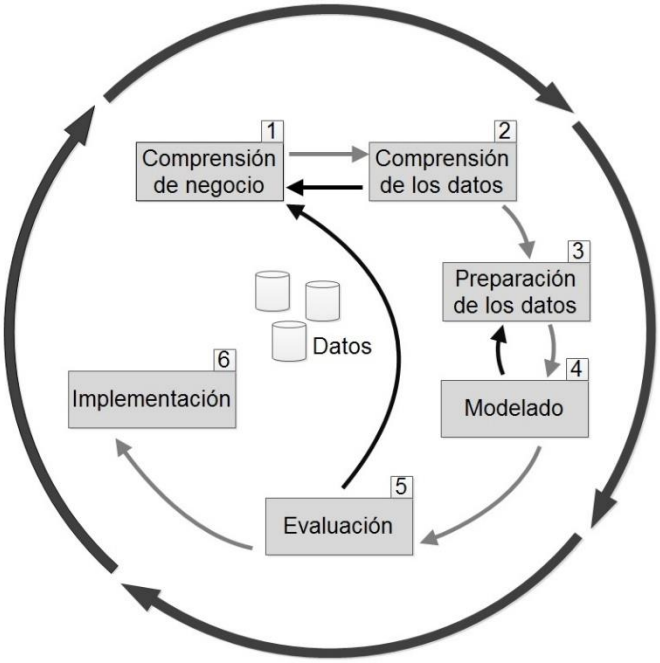

Figura. 9 Fases del Ciclo de Vida de CRISP-DM

El proceso es dinámico e iterativo, por lo que la ejecución de los procesos no es estricta y con frecuencia se puede pasar de uno a otro proceso, de atrás hacia delante y viceversa. Estos dependen del resultado de cada fase o la planeación de la siguiente tarea por ejecutar.

A continuación se describe cada una de las seis fases de la metodología CRISP-DM.

3.2.2 Fase 1. Comprensión del negocio o problema.

En esta fase, se define los objetivos del proyecto y se plantea los requisitos del mismo, tomando en cuenta la perspectiva del negocio; con esta información se define el problema a resolver y se genera el plan del proyecto de minería de datos.

Esta fase está compuesta por cuatro tareas específicas, las cuales se describen a continuación:

1. Determinación de objetivos de negocio; en esta tarea el analista de datos debe entender desde la perspectiva del negocio lo que el cliente quiere lograr. Muchas veces el cliente puede tener diferentes objetivos y restricciones que

compiten entre ellos, los cuales deben ser equilibrados (Chapman, y otros, 2000). Las salidas de esta tarea son:

   a. Objetivos del negocio: Se describe el objetivo primario del cliente desde una perspectiva del negocio.
   b. Criterios de éxito del negocio: Se describe los criterios para un resultado útil del proyecto.

2. Evaluación de la Situación; "esta tarea implica la investigación más detallada sobre todos los recursos, restricciones, presunciones, y otros factores que deberían ser considerados en la determinación del objetivo de análisis de datos y el plan de proyecto" (Chapman, y otros, 2000). Las salidas de esta tarea son:
   a. Inventario de recursos: Se lista los recursos disponibles para el proyecto, incluyendo personal, datos, hardware y software.
   b. Requerimientos, asunciones y restricciones: Se lista todos los requerimientos del proyecto, presunciones sobre los datos o negocio, restricciones sobre la disponibilidad de recursos, tecnología y acceso a datos.
   c. Riesgos y contingencias: Se lista los eventos que pueden retrasar el proyecto o hacer que falle y las correspondientes acciones a realizar en caso de su ocurrencia.
   d. Terminología: Se debe incluir un glosario de términos del proyecto y otro de minería de datos.
   e. Costos y beneficios: Se desarrolla un análisis costo-beneficio para el proyecto.
3. Determinación de los objetivos de la minería de datos; se determina los objetivos del proyecto de forma técnica (Chapman, y otros, 2000). Las salidas de esta tarea son:
   a. Objetivos de la minería de datos
   b. Criterios de éxito de la minería de datos
4. Desarrollo del Plan de Proyecto; para esta tarea se puede utilizar una plantilla de definición del "*Plan de Proyecto*" de acuerdo a las recomendaciones del PMI. El plan debe contener todos los objetivos especificados en las tareas anteriores.

La salida de esta tarea es el documento de Plan de Proyecto (Chapman, y otros, 2000).

3.2.3 Fase 2. Comprensión de los datos

Esta fase comprende la recolección inicial de datos, con el objetivo de establecer un primer contacto con el problema, familiarizándose con ellos, identificar su calidad y establecer las relaciones más evidentes que permitan definir las primeras hipótesis. Es muy común en ciertos proyectos la necesidad de crear nuevos repositorios de datos específicamente para el proyecto de minería de datos, ya que el continuo acceso y consulta a los datos puede ocasionar demoras o bloqueos en el sistema de producción de la empresa (Arancibia, 2009).

Esta fase está compuesta por cuatro tareas específicas, las cuales se describen a continuación:

1. Recolección de datos iniciales; en esta tarea se elaboran informes con una lista de los datos disponibles, su localización, las técnicas utilizadas en su recolección y los problemas y soluciones del proceso (Arancibia, 2009). La salida de esta tarea es el "*Informe Inicial de Recolección de Datos*" en el que se describe los datos obtenidos, los métodos usados para obtenerlos y los problemas que se pudieron encontrar.
2. Describir los datos; en esta tarea se debe describir de forma general los datos obtenidos. La salida de esta tarea es el "*Informe de Descripción de Datos*" en el que se describe los datos obtenidos, el formato y cantidad de los mismos (Chapman, y otros, 2000).
3. Explorar los Datos; "esta tarea dirige interrogantes de minería de datos usando preguntas, visualización, y técnicas de reporte" (Chapman, y otros, 2000). La salida de esta tarea es el "I*nforme de Exploración de Datos*" en el que se incluyen las primeras conclusiones o hipótesis sobre los datos obtenidos.
4. Verificar la calidad de los datos; en esta tarea, se efectúan verificaciones sobre los datos para determinar la consistencia de los valores de los campos, la cantidad y distribución de los valores nulos, y encontrar valores fuera de rango. El objetivo de esta tarea es asegurar la completitud y corrección de los datos

(Arancibia, 2009). Como salida de esta tarea se elabora un informe de la verificación realizada.

3.2.4 Fase 3. Preparación de los datos

En esta fase se procede a la preparación de los datos para adaptarlos a las técnicas de minería de datos que se van a utilizar. La preparación de datos incluye las tareas de selección de datos a los que se va a aplicar una determinada técnica de modelado, limpieza de datos, generación de variables adicionales, integración de diferentes orígenes de datos y cambios de formato (Arancibia, 2009). La principal salida de esta fase es el *"Conjunto de datos"* que será utilizado para producir el modelo del proyecto de minería de datos (Chapman, y otros, 2000).

Esta fase está compuesta por cinco tareas específicas, las cuales se describen a continuación:

1. Selección de datos: en esta tarea se selecciona un subconjunto de los datos adquiridos para su posterior análisis. Dependiendo de los datos que se dispone y de los objetivos del proyecto, así como de la calidad, volumen y restricciones se selecciona los datos que serán analizados y utilizados en la construcción del modelo (Chapman, y otros, 2000). La salida de esta tarea es la lista de "*Razonamiento para la inclusión/exclusión"* el cual detalla además las razones de la inclusión o exclusión de los datos.
2. Limpieza de los datos: esta tarea complementa a la anterior, y es una de las que más tiempo y esfuerzo consume, debido a la diversidad de técnicas que pueden aplicarse para optimizar la calidad de los datos. Algunas de las técnicas a utilizar para este propósito son: normalización de los datos, discretización de campos numéricos, tratamiento de valores ausentes y reducción del volumen de datos (Arancibia, 2009).
3. Estructuración de los datos: en esta tarea se realizan operaciones de preparación de los datos como: generación de nuevos atributos, integración de nuevos registros o transformación de valores para atributos existentes (Arancibia, 2009).

4. Integración de los datos: esta tarea involucra la creación de nuevas estructuras a partir de los datos seleccionados, creación de nuevos registros, fusión de tablas campos o nuevas tablas donde se resumen características de múltiples registros (Arancibia, 2009).
5. Formateo de los datos: en esta tarea se ejecutan transformaciones sintácticas de los datos sin modificar su significado, esto, con la idea de permitir o facilitar el empleo de alguna técnica de minería de datos en particular, las tareas comunes son la reordenación de los campos y/o registros, el ajuste de valores de los campos por las posibles limitaciones de las herramientas de modelado (Arancibia, 2009).

3.2.5 Fase 4. Modelado

En esta fase se selecciona las técnicas de modelado más apropiadas para el desarrollo del proyecto de minería de datos que se está desarrollando. La selección debe basarse en los siguientes criterios (Arancibia, 2009):

- Ser apropiada al problema
- Disposición de los datos adecuados
- Cumplimiento de los requisitos del proyecto
- Conocimiento de la técnica

Esta fase está compuesta por cuatro tareas específicas, las cuales se describen a continuación:

1. Selección de la técnica de modelado: en esta tarea se selecciona la técnica de minería de datos más apropiada al tipo de problema a resolver. Para esta selección, se debe considerar el objetivo principal del proyecto y la relación con las herramientas existentes (Chapman, y otros, 2000).
2. Generación del plan de prueba: en esta tarea se debe generar un procedimiento para probar la calidad y validez del modelo que se va a construir. Por lo general se separan los datos en dos conjuntos, uno de entrenamiento y otro de prueba, para luego construir el modelo basado en el

conjunto de entrenamiento y medir la calidad del modelo generado con el conjunto de prueba (Arancibia, 2009).

3. Construcción del Modelo. Esta tarea consiste en la aplicación de la técnica de minería de datos seleccionada sobre los datos previamente preparados para generar uno o más modelos. Adicionalmente, se debe verificar los valores de los diferentes parámetros de la técnica seleccionada y en un proceso iterativo ajustar los valores de los mismos de acuerdo a los resultados obtenidos en la siguiente tarea *"Evaluación del Modelo"* (Arancibia, 2009).
4. Evaluación del modelo. En esta tarea se interpretan los modelos de acuerdo al conocimiento preexistente del dominio y los criterios de éxito preestablecidos. Expertos en el dominio del problema juzgan los modelos dentro del contexto del dominio (Arancibia, 2009).

3.2.6 Fase 5. Evaluación

Esta fase evalúa el grado al que el modelo responde a los objetivos de negocio, teniendo en cuenta el cumplimiento de los criterios de éxito del problema y procura determinar si hay alguna decisión de negocio por el que este modelo es deficiente. Es preciso revisar el proceso, en base a los resultados obtenidos para poder repetir algún paso anterior, en el caso de que se haya cometido algún error (Chapman, et al., 2000).

Esta fase está compuesta por tres tareas específicas, las cuales se describen a continuación:

1. Evaluación de los resultados: en esta tarea involucra la evaluación del modelo en relación a los objetivos del negocio. Además de los resultados directamente relacionados con el objetivo del proyecto, también podría revelar desafíos adicionales, información, o insinuaciones para futuras direcciones (Chapman, y otros, 2000).
2. Proceso de revisión. El proceso de revisión, se refiere a calificar al proceso entero de minería de datos ejecutado, para identificar elementos que pudieran ser mejorados y determinar si hay algún factor importante o tarea que pudo haber sido pasada por alto (Arancibia, 2009).

3. Determinación de futuras fases. Si se ha determinado que las fases hasta este momento han generado resultados satisfactorios, podría pasarse a la implementación, en caso contrario puede decidirse realizar otra iteración desde la fase de preparación de datos o de modelación con otros parámetros algunas veces incluso puede decidirse partir de cero y realizar un nuevo proyecto (Chapman, y otros, 2000).

3.2.7 Fase 6. implementación

En esta fase se transforma el conocimiento obtenido en acciones dentro del proceso de negocio. Un proyecto de minería de datos no concluye en la implantación del modelo, pues se deben documentar y presentar los resultados de manera comprensible para el usuario, con el objetivo de lograr un incremento del conocimiento. Además se debe asegurar el mantenimiento de la aplicación y el uso de los resultados obtenidos (Arancibia, 2009).

Esta fase está compuesta por cuatro tareas específicas, las cuales se describen a continuación:

1. Plan de Despliegue; esta tarea toma los resultados de la evaluación y concluye una estrategia para su implementación en los procesos del negocio (Chapman, y otros, 2000).
2. Plan de Monitoreo y Mantenimiento; cuando los modelos resultantes de la ejecución del proyecto son "*puestos en producción";* es decir son incorporados en el trabajo diario de la organización o en sus procesos es necesario contar con un plan detallado de supervisión de los mismos (Chapman, y otros, 2000).
3. Informe Final; esta tarea consiste en la elaboración de un informe final del proyecto, el cual generalmente es un resumen del proyecto y las experiencias logradas, también puede ser una presentación final que incluya y explique los resultados logrados con la ejecución del proyecto (Arancibia, 2009).
4. Revisión del Proyecto; en esta tarea se evalúa lo que se hizo bien, y lo que se hizo mal o los errores que se cometieron, como experiencias a ser

utilizadas y tomadas en cuenta en futuros proyectos (Chapman, y otros, 2000).

El documento “CRISP-DM 1.0 Step-by-step data mining guide” elaborada por (Chapman, y otros, 2000); además de la explicación y del detalle de la metodología expuesta contiene una “Guía del Usuario”; la cual facilita la comprensión de la aplicación de la misma.

Finalmente es bueno aclarar que desde que la metodología fue publicada en el año 2000, la misma no ha tenido nuevas versiones; por lo que el conocimiento y uso de la misma se ha popularizado entre los investigadores de datos.

CAPITULO IV
DESARROLLO DE LA INVESTIGACION

4.1 Introducción

La empresa Gestor, es una empresa que cuenta con 19 años de vida en la industria del software local. Gestor basa su giro de negocio en la venta de su producto estrella "Gestor Web Fiducia Fondos", el mismo que ha evolucionado lo que ha permitido su comercialización en 20 países de toda Latinoamérica; en la actualidad Gestor está asumiendo nuevos retos incorporando a su plan de negocio la venta de servicios de consultoría especializada en el ámbito fiduciario y de la banca de inversión; además de aprovechar los nuevos esquemas de negocio como la nube y la comercialización de su producto bajo los esquemas SaaS y PaaS.

Gestor tiene dos grupos técnicos para el desarrollo de su operación; el primero llamado "Core" es el encargado de evolucionar el producto "Gestor Web Fiducia Fondos" añadiendo nuevas funcionalidades; corrigiendo errores y publicando actualizaciones para los clientes que deseen incorporarlas a su producto; el segundo grupo es el de "Proyectos", este grupo cuenta con profesionales certificados en dirección de proyectos, expertos funcionales del producto y desarrolladores; quienes son los encargados de los proyectos de implantación.

Por las características del producto "Gestor Fiducia Fondos" que se especializa en la gestión de fideicomisos y fondos de pensiones dando soporte a la mayoría de legislaciones de los países de Latinoamérica; la comercialización del mismo lleva períodos de seis meses en el mejor de los casos hasta dos años; similar a lo que implica la comercialización de sistemas especializados como ERP, CRM o Core Bancarios.

En cada venta; el cliente además de pagar por la licencia de uso del sistema debe invertir en la compra de infraestructura tecnológica (servidores, elementos de red como swtich, firewall) y de la plataforma de software (licencias de productos Oracle, base de datos y plataforma Forms & Reports) necesarias para el despliegue del producto. Además de esta inversión inicial el cliente debe contratar el proyecto de

implantación del producto; en el que se realizan modificaciones al mismo para cubrir las necesidades específicas que tiene cada cliente.

Las instituciones financieras que desean adquirir un producto de software especializado para optimizar su operación, publican concursos internacionales para que las casas de software presenten sus ofertas, estos concursos se basan en documentos de alto nivel llamados “RFP” (Solicitud de Propuesta) los cuales contienen las necesidades puntuales de los clientes especificadas a alto nivel. En base a estos documentos la empresa que desea participar en la licitación; debe analizar si su producto cumple con todas las necesidades del cliente y adicionalmente calcular el tiempo y costo necesario para desarrollar las funcionalidades con las que no cuenta el producto o modificar las que necesitan ser ajustadas; uno de los puntos que Gestor desea mejorar es la estimación del esfuerzo para estas adiciones o cambios funcionales a su producto; que le permitan incrementar sus ganancias y reducir el riesgo.

Para satisfacer esta necesidad actual de la empresa Gestor se ha planteado la necesidad de la realización de un proyecto de minería de datos; que permita platear un modelo que facilite la estimación del esfuerzo necesario para el desarrollo de las necesidades de sus clientes en la etapa de elaboración de la propuesta sin contar con requerimientos detallados.

4.2 Ejecución del Proyecto de Minería de Datos

Para la ejecución del proyecto de minería de datos; producto de la investigación de este trabajo de titulación se ha seleccionado la metodología CRISP-DM, la cual se explicó a detalle en el capítulo anterior. A continuación se describe como se ejecutó cada una de las fases propuestas por la metodología para obtener al final un modelo de redes neuronales artificiales que satisface las necesidades de la empresa.

4.2.1 Fase 1 - Comprensión del negocio o problema

Determinación del problema a resolver

De acuerdo a lo explicado en la introducción de este capítulo, el problema que desea resolver Gestor, es el de mejorar la precisión de la estimación del esfuerzo requerido para el desarrollo de nuevas funcionalidades o adaptación de las actuales de su producto “Gestor Web Fiducia Fondos” en base a la especificación de requerimientos de alto nivel proporcionados por sus potenciales clientes.

La resolución de este problema reducirá el riesgo en todas las etapas del proyecto en caso de ganar una licitación de implantación de su producto; adicionalmente permitirá optimizar sus recursos; al conocer con anticipación el esfuerzo necesario y poder direccionar los esfuerzos desde el inicio del proyecto y no a medida en que la probabilidad de retraso en el proyecto aumenta por estimaciones iniciales demasiado optimistas.

Criterios de Éxito

La fiabilidad del modelo definido como producto del proyecto de minería de datos estará expresada en los siguientes factores que se explican a continuación:

- Confiabilidad del modelo, comparando los resultados obtenidos de este con la duración real de proyectos anteriormente ejecutados por la empresa.
- Permitir identificar debilidades y falencias en el proceso de desarrollo actual; en base al análisis de la información histórica que mantiene la empresa.
- Definir nuevos atributos que deberían ser considerados para mejorar la precisión de las estimaciones y que actualmente la empresa no toma en cuenta en la información registrada por sus colaboradores.

Evaluación de Antecedentes y Requisitos

En base a las necesidades de la organización y una revisión previa de la información que dispone para la ejecución de este proyecto se menciona los antecedentes que se debe tener en cuenta al momento de realizar la investigación;

así como los requisitos con los que debe cumplir el proyecto para satisfacer las necesidades de Gestor.

Antecedentes

- La empresa cuenta con información detallada del esfuerzo real en el desarrollo de requisitos de software sobre su producto “Gestor Web Fiducia Fondos” de diferentes proyectos de implantación realizados en los últimos 8 años.
- La empresa actualmente cuenta con tres diferentes productos: “Gestor Web Fiducia Fondos”, “Gestor G5 Trust” y “Gestor Banca Electrónica”; los dos últimos productos se encuentran en fase de implantación y pruebas con un solo cliente; y están desarrollados bajo los estándares de la plataforma “*Java Enterprise Edition (JEE)*”[2] por lo que no serán considerados en la definición del modelo de estimación.
- A pesar de tener información detallada la empresa no ha realizado análisis estadísticos profundos en sus datos, por lo que considera que esta es la oportunidad ideal para descubrir información que ha permanecido oculta, y la deducen solo de forma empírica.
- Actualmente Gestor, realiza la estimación del esfuerzo utilizando el método de Juicio Experto, apoyado por una hoja de cálculo que simplemente da pesos preestablecidos en base a una identificación de la complejidad del requisito.

Requisitos

- El modelo de estimación, deberá ser comprobado con la información histórica para validar su grado de precisión
- Se utilizará la información que actualmente se encuentra registrada en el sistema “ISOv2”, propiedad de Gestor, el cual se ha utilizado en los últimos 10 años para registrar los tiempos de sus colaboradores en el proceso de desarrollo.

[2] Es una plataforma de programación para desarrollar y ejecutar aplicaciones de software en el lenguaje Java; Permite utilizar arquitecturas de N capas distribuidas las que se ejecutan en un servidor de aplicaciones. http://www.oracle.com/technetwork/java/javaee/overview/index.html

- Se deberá tomar en cuenta la información de los últimos 5 años; ya que la información anterior corresponde a un proceso anterior que era manejado por la empresa y la información no es fiable.
- El producto para el cuál se desea elaborar el modelo de estimación es “Gestor Web Fiducia Fondos”; la información relacionada a proyectos de otros productos de la empresa no será tomada en cuenta.

Plan de Proyecto

El plan del proyecto, se encuentra en el Anexo 1, de este documento de titulación, para el mismo se ha utilizado una plantilla que sigue las recomendaciones del PMI y ha sido adaptada a las necesidades de Gestor.

4.2.2 Fase 2 - Comprensión de los datos

Para está fase se ha estudiado la información contenida en el sistema ISOv2, así como la estructura del modelo entidad-relación del mismo. Una ventaja que vale la pena mencionar es que se cuenta con el conocimiento completo de la información almacenada así como de la estructura de la misma ya que ISOv2 es una herramienta propia de Gestor.

Recolección de Datos

El sistema ISOv2, es un sistema desarrollado bajo una arquitectura tradicional Cliente-Servidor; que se despliega bajo los lineamientos de la plataforma “Oracle Forms&Reports”[3]. El sistema cuenta con una base de datos entidad relación en la cual se almacena toda la información del mismo. La ***Figura 10***, detalla la arquitectura del sistema ISOv2.

[3] Oracle Forms & Reports es una plataforma de software para la creación de aplicaciones Cliente/Servidor orientadas al usuario final que interactúan con una base de datos Oracle. https://docs.oracle.com/en/middleware/

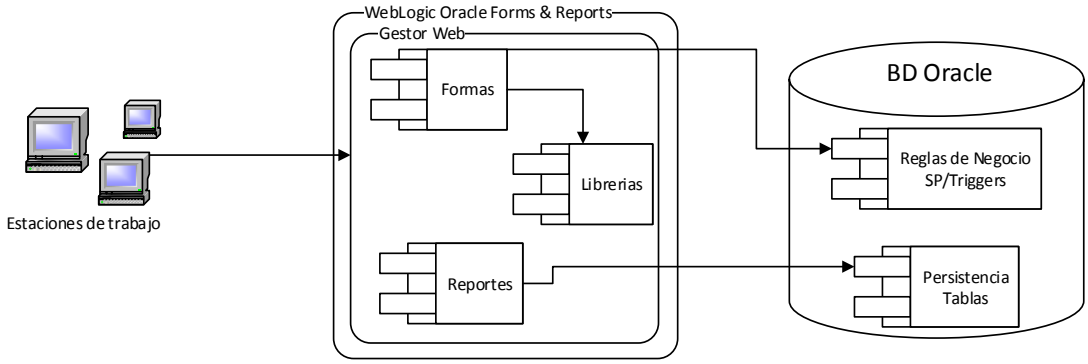

Figura 10 Arquitectura sistema Gestor ISOv2

El sistema ISOv2; cuenta con las siguientes funcionalidades: Gestión de Usuarios, Gestión de Personal, Gestión de Presupuesto y Gestión de Proyectos. Esta última funcionalidad incluye las siguientes características: Gestión de Clientes, Gestión de Requisitos y Aseguramiento de la Calidad del producto.

En base a la descripción anterior; se limita el ámbito de recolección de datos al esquema de base de datos, del sistema ISOv2; utilizando las estructuras de información (tablas) que forman parte de la funcionalidad: Gestión de Proyectos.

Analizando el esquema de la base de datos, se encuentra que las estructuras de información relacionadas con la funcionalidad Gestión de Proyectos, son 11; estas tablas dan soporte a todas las características de la funcionalidad por lo que es necesario realizar una selección de las tablas que contienen la información necesaria para la construcción del modelo. La tabla 1, detalla las tablas de la funcionalidad Gestión de Proyectos, una breve descripción de cada una y finalmente un atributo que indica si la tabla será tomada en cuenta en base a la información que almacena.

Tabla 1
Entidades de la Funcionalidad Gestión de Proyectos

TABLA	DESCRIPCIÓN	SELE.
FD_USUARIO	Dentro de esta tabla se almacenarán todos los usuario del Sistema ISOv2	SI
ISO_CLASE_COMUNICACION	Clasificador secundario para los tipos de comunicación	NO
ISO_HISTORICO_TIEMPO_EJECUTADO	Dentro de esta tabla se almacena el detalle de tiempo dedicado al cumplimiento de una orden de trabajo.	SI
ISO_ORDEN_CONTROL_CALIDAD	Dentro de esta tabla se almacenan las ordenes de trabajo asignadas a los responsables de Control de Calidad para su proceso de control, almacena los números de errores por cada tipo y si está liberadas cada una de las instancias de control	SI
ISO_ORDEN_TRABAJO	Dentro de esta tabla se almacenan las órdenes de trabajo para los responsables de actividades dentro de la Compañía.	SI
ISO_PRODUCTO_MODULO	Módulos dentro de cada Producto	NO
ISO_PROYECTO	Dentro de esta tabla se almacenarán los proyectos sobre los cuales se realizarán actividades dentro de la Compañía.	SI
ISO_PROYECTO_FASE	Fases dentro del Proyecto	NO
ISO_REG_MODIFICA_MEJORAS	Dentro de esta tabla se almacenarán los objetos modificados o creados durante el proceso de desarrollo de una determinada Orden de trabajo.	SI
ISO_RELOJ_CONTROL	Almacena el tiempo que se toma el realizar el proceso de revisión de órdenes de trabajo	SI
ISO_TIPO_COMUNICACION	Dentro de esta tabla se almacenarán los tipos de comunicación que indicarán el comportamiento de las comunicaciones en todo su proceso.	NO

Una vez que se ha seleccionado las tablas que contienen la información de utilidad con el proyecto se ha procedido a importarlas a una herramienta de modelado para comprender y visualizar de mejor forma las “relaciones” que tienen entre ellas. El término relaciones hace referencia a las relaciones de un modelo entidad-relación, nivel físico. La ***Figura. 11***, muestra el modelo “Entidad-Relación” generado por la herramienta de modelamiento del conjunto de tablas seleccionadas.

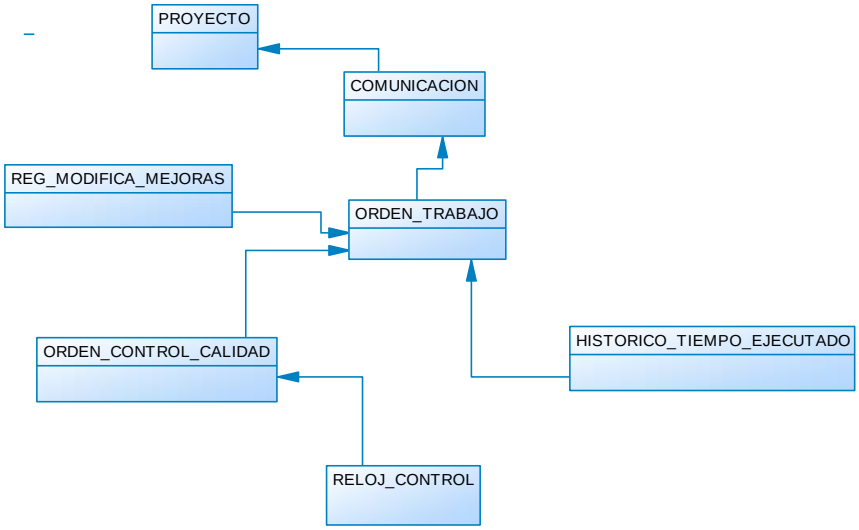

Figura. 11 Modelo Entidad-Relación parcial del sistema ISO V2

Descripción de los Datos

Una vez que se ha seleccionado las tablas y se ha entendido la relación existente entre las mismas, se ha procedido a investigar y detallar la estructura de cada tabla para validar su uso. A continuación se muestra la estructura de cada una de las entidades seleccionadas; esta descripción se la ha realizado en múltiples tablas que contienen la siguiente información: nombre del campo, tipo de dato y descripción; de esta forma se facilita la comprensión de cada una de las estructuras. Es necesario aclarar que no se incluyen todas las columnas de las tablas, sino solo las columnas seleccionadas, ya que cada tabla cuenta en promedio con 3 columnas de relaciones, y 8 columnas de auditoria; las cuales no proveen información útil para el modelo que se busca construir y su inclusión generaría más confusión antes que agregar valor a la demostración de lo realizado.

Tabla 2

Descripción de la tabla: ISO_PROYECTO

Nombre Campo	Tipo Dato	Descripción
cod_proyecto	varchar2(8)	código del proyecto
nombre	varchar2(100)	nombre del proyecto
descripcion	varchar2(4000)	descripción del proyecto
cod_cliente	varchar2(30)	código del cliente del proyecto
cod_responsable	varchar2(30)	código del responsable del proyecto

Tabla 3

Descripción de la tabla: ISO_COMUNICACION

Nombre Campo	Tipo Dato	Descripción
nro_comunicacion	number(8)	secuencial único que se asignará de manera automática a cada registro de comunicación
cod_producto	varchar2(8)	código del producto sobre el cual se registra la comunicación
cod_proyecto	varchar2(30)	código del proyecto dentro del cual se registra la comunicación
cod_fase	varchar2(8)	fase dentro del proyecto al cual pertenecerá la comunicación
fecha_inicio	date	fecha estimada de inicio de proceso de la comunicación
cod_modulo	varchar2(8)	código del módulo del producto para el cual se registra la comunicación
cod_responsable	varchar2(30)	código del responsable de atención al cliente para el registro de la comunicación
cod_tipo_comunicacion	varchar2(8)	tipo de comunicación que se registre, tendrá un efecto en base a las parametrización de cada uno de los tipos de comunicación
cod_prioridad	varchar2(8)	prioridad asignada a cada comunicación
cod_estado_comunicacion	varchar2(3)	estado en el cual se encuentra la comunicación (pen, pendiente, fin finalizado, rec rechazado, anu anulado)
tiempo_horas	number(8,2)	Tiempo estimado en horas que se planifica para la comunicación.

Tabla 4

Descripción de la tabla: ISO_ORDEN_TRABAJO

Nombre Campo	Tipo Dato	Descripción
nro_comunicacion	number(8)	Corresponde al número de comunicación registrado.
nro_orden	number(10)	Secuencial único para identificar las ordenes.
tiempo_estimado	number(8,2)	indica el tiempo estimado de la orden, en caso que no implique desarrollo será manual, caso contrario será automático
fecha_inicio	date	fecha en la que el responsable indica que se inicia el desarrollo
fecha_final	date	fecha en la que el responsable indica que se finaliza el desarrollo
responsable	varchar2(30)	Responsable de desarrollar la orden.
tiempo_estimado_analisis	number(8,2)	indica el tiempo que se estima de manera automática para el análisis del desarrollo
tiempo_estimado_desarrollo	number(8,2)	indica el tiempo que se estima de manera automática para el desarrollo de la orden
dias_adicionales_orden_calidad	number(3)	indica un tiempo de prorroga que se aplicará a las órdenes que calidad generó, el tiempo de prorroga se restará del tiempo transcurrido y se lo comprará con el parámetro registrado en la tabla iso_parametros_generales
complejidad	number(1)	indica la complejidad de la orden (4 muy alta, 3 alta, 2 media, 1 baja)

Tabla 5

Descripción de la tabla: ISO_HISTORICO_TIEMPO_EJECUTADO

Nombre Campo	Tipo Dato	Descripción
nro_orden	number(10)	número de la orden a la cual se registrará el tiempo dedicado a su cumplimiento
secuencia	number(5)	Número secuencial discriminador que permite tener varios registros de tiempo para una misma orden.
nro_comunicacion	number(8)	Número de comunicación de cliente a la cual pertenece la orden.
fecha	date	Fecha para la cual se registra el tiempo.
tiempo_horas	number(8,2)	Tiempo dedicado al cumplimiento de la orden en horas.
nro_revision	number(2)	Número. de revisión sobre la cual está aplicado del tiempo.
porcentaje_avance	number(6,2)	Porcentaje de avance que representa el tiempo invertido en la orden.
nro_revision_supervisor	number(2)	Corresponde al número de revisión de supervisor o implantador.
genera_calidad	varchar2(1)	indica si el registro fué creado por calidad

Tabla 6

Descripción de la tabla: ISO_REG_MODIFICA_MEJORAS

Nombre Campo	Tipo Dato	Descripción
nro_orden	number(10)	Número de orden de trabajo
origen	varchar2(2)	indica el origen del objeto (bd: base de datos ar: archivo)
tipo_bd	varchar2(20)	si se refiere a un objeto de base de datos indica el tipo de objeto de base de datos(procedure, function, trigger, etc)
estado	varchar2(3)	Indica el estado del objeto dentro del registro act: activo, ina: inactivo.
es_objeto	varchar2(1)	Indica si es un objeto de la aplicación.

Tabla 7

Descripción de la tabla: ISO_ORDEN_CONTROL_CALIDAD

Nombre Campo	Tipo Dato	Descripción
nro_orden	number(10)	Número de la orden sobre la cual se realizará control de calidad
nro_revision	number(2)	Número de revisión que corresponde al control de calidad
nro_comunicacion	number(8)	Número de comunicación dentro de la cual se encuentra registrada la orden.
fecha_inicio	date	fecha en la cual se inicia el control de calidad
fecha_fin	date	fecha en la cual se finaliza el control de calidad
estado	varchar2(3)	estado en el cual se encuentra el control de calidad (pen pendiente, pos pospuesto, rev en revisión, fin finalizado)

Tabla 8

Descripción de la tabla: ISO_RELOJ_CONTROL

Nombre Campo	Tipo Dato	Descripción
usuario	varchar2(30)	código identificador único para el usuario
secuencia	number(20)	número secuencial de la revisión
nro_orden	number(10)	Número de la orden sobre la cual se realizará control de calidad
nro_revision	number(2)	Número de revisión que corresponde al control de calidad
fecha_inicio	date	fecha en la cual se inicia el control de calidad
fecha_fin	date	fecha en la cual se finaliza el control de calidad

Tabla 9

Descripción de la tabla: FD_USUARIO

Nombre Campo	Tipo Dato	Descripción
cod_usuario	varchar2(30)	código identificador único para el usuario
nombre	varchar2(100)	nombre del usuario

Exploración de Datos

Para facilitar la tarea de exploración de los datos se ha creado una vista materializada, la cual nos permite la exploración de los datos en una tabla de dos dimensiones; de esta forma la visualización inicial de los datos que pasaran a las

siguientes fases del proceso se facilita en gran manera. La instrucción SQL utilizada para la construcción de la tabla materializada se la muestra a continuación:

```
CREATE OR REPLACE VIEW ISO V QA CONTROL ORDEN AS
select a.nro Comunicacion,a.nro orden,b.cod tipo comunicacion,c.nombre
nombre_tipo_comunicacion,cod_estado_orden,b.cod_prioridad,b.cod_cliente,
       c.cod_grupo,d.nombre nombre_grupo,c.cod_clase,e.nombre Nombre_Clase,
       a.descripcion,a.resumen,trunc(a.fecha_inicio)
Fecha inicio planificada,trunc(a.fecha final) Fecha final Planificada,
       a.tiempo estimado,a.fecha finaliza orden,a.fecha aprueba,b.cod proyecto,h.nombr
e nombre proyecto,b.cod fase,f.nombre Nombre fase,b.cod modulo,g.nombre nombre modulo,
       a.cod_estado_orden
Estado_orden,a.es_reproceso,f_obt_qa_estado_Orden(a.nro_orden,'PAR')
Prueba_Par,f_obt_qa_estado_Orden(a.nro_orden,'PU') Pruebas_unitarias,
       b.estado calidad Estado validacion,a.responsable
Responsable Orden,a.cod unidad,a.cod subunidad,b.cod responsable
Responsable_Comunicacion,
       (select NVL(sum(y.porcentaje_Avance),0) from iso_historico_tiempo_ejecutado
y where y.nro_orden=a.nro_orden) porcentaje_avance,
       a.cod responsable prueba par,
       b.cod responsable valida,b.cod estado comunicacion,b.cod requerimiento,a.fecha
solicita_prueba_par,a.fecha_aprueba_par,
       (select min(y.fecha) from iso_historico_tiempo_ejecutado
y where y.nro_orden=a.nro_orden) Fecha_ini_reg_tiempo,
       (select max(z.fecha) from iso_historico_tiempo_ejecutado
z where z.nro orden=a.nro orden) Fecha fin reg tiempo,
       (select max(x.nro revision) from iso orden revision
x where x.nro_orden=a.nro_orden) Nro_revision_par,
       (select max(x.nro_revision) from iso_orden_control_calidad
x where x.nro_orden=a.nro_orden) Nro_revision_pu,
       (select x.responsable_control from iso_orden_control_calidad
x where x.nro orden=a.nro orden
        and nro revision=(select max(x.nro revision) from iso orden control calidad
x where x.nro_orden=a.nro_orden)) Responsable_PU,
       (select x.fecha_asigna from iso_orden_control_calidad
x where x.nro_orden=a.nro_orden
        and nro_revision=(select max(x.nro_revision) from iso_orden_control_calidad
x where x.nro orden=a.nro orden)) Fecha Asignacion pu,
       (select x.fecha_inicio from iso_orden_control_calidad
x where x.nro_orden=a.nro_orden
        and nro revision=(select max(x.nro revision) from iso orden control calidad
x where x.nro orden=a.nro orden)) Fecha Revision pu,
       (select x.fecha fin from iso orden control calidad
x where x.nro orden=a.nro orden
        and nro_revision=(select max(x.nro_revision) from iso_orden_control_calidad
x where x.nro_orden=a.nro_orden)) Fecha_Finaliza_pu,
       (select sum(x.tiempo horas) from iso historico tiempo ejecutado
x where x.nro orden=a.nro orden) Horas orden,
       (select sum(round((x.fecha fin-
x.fecha inicio)*24,2)) from iso orden revision reloj
x where x.nro_orden=a.nro_orden and x.fecha_fin is not null) Tiempo_revision_par,
       (select sum(round((x.fecha_fin-x.fecha_inicio)*24,2)) from iso_reloj_control
x where x.nro orden=a.nro orden and x.fecha fin is not null) Tiempo revision pu,
       (select sum(x.tiempo horas) from iso historico tiempo ejecutado
x where x.nro orden=a.nro orden and (x.nro revision!=1 or x.es reproceso='S'))
Horas orden reproceso,
       (select sum(round((x.fecha_fin-
x.fecha_inicio)*24,2)) from iso_orden_revision_reloj
x where x.nro orden=a.nro orden and x.fecha fin is not null and (x.nro revision!=1 or
x.es reproceso='S')) Tiempo revision par reproceso,
       (select sum(round((x.fecha fin-x.fecha inicio)*24,2)) from iso reloj control
x where x.nro_orden=a.nro_orden and x.fecha_fin is not null and (x.nro_revision!=1 or
x.es_reproceso='S')) Tiempo_revision_pu_reproceso,
       (select x.fecha solicita revision from iso comunicacion revision
x where x.nro comunicacion=a.nro comunicacion
        and nro revision=(select max(x.nro revision) from iso comunicacion revision
x where x.nro_comunicacion=a.nro_comunicacion)) Fecha_solicita_validacion,
       (select x.fecha_finaliza_revision from iso_comunicacion_revision
x where x.nro_comunicacion=a.nro_comunicacion
```

```
        and nro_revision=(select max(x.nro_revision) from iso_comunicacion_revision
x where x.nro_comunicacion=a.nro_comunicacion)) Fecha_finaliza_validacion,
       (select max(x.nro_revision) from iso_comunicacion_revision
x where x.nro_comunicacion=a.nro_comunicacion) Nro_revision_validacion,
       (select sum(round((x.fecha fin-
x.fecha inicio)*24,2)) from iso comunicacion revis reloj
x where x.nro_comunicacion=a.nro_comunicacion and x.fecha_fin is not null)
Tiempo_revision_Validacion,
       (select decode(count(x.num_incidente),0,'N','S') from iso_incidentes
x where x.nro comunicacion rel=a.nro comunicacion) es incidente,
       (select x.num incidente from iso incidentes
x where x.nro_comunicacion_rel=a.nro_comunicacion) num_incidente,
       (SELECT LISTAGG(x.num_ticket, ' ') WITHIN GROUP  (ORDER BY x.num_ticket)
        FROM   iso_comunicacion_sdg_ticket x
        where  b.nro_comunicacion=x.nro_comunicacion) Ticket,
       nvl((select n.nombre from iso agrupador tipo comunicac m,iso agrupador tipo n
        where n.cod agrupador=m.cod agrupador
        and   m.cod_unidad=a.cod_unidad
        and   m.cod_tipo_comunicacion=b.cod_tipo_comunicacion),'NO RELACIONADO')
Agrupador Prod
from   iso orden trabajo a,iso Comunicacion b,iso tipo comunicacion
c,iso grupo comunicacion d,iso clase comunicacion e,iso proyecto fase f,
       iso_producto_modulo g,iso_proyecto h
where  a.nro_comunicacion=b.nro_comunicacion
and    b.cod_tipo_comunicacion=c.cod_tipo_comunicacion
and    c.cod grupo=d.cod grupo
and    c.cod clase=e.cod clase
and    b.cod proyecto=f.cod proyecto
and    b.cod_fase=f.cod_fase
and    b.cod_producto=g.cod_producto
and    b.cod_modulo=g.cod_modulo
and    b.cod proyecto=h.cod proyecto
and    trunc(a.fecha_inicio)>'01-01-2010';
```

Es bueno aclarar, que la construcción de esta tabla materializada fue un proceso iterativo, ya que las necesidades de información se dieron en el transcurso del proyecto; sin embargo, aquí solo se muestra la instrucción final. A continuación se muestra la ***Figura 12***, la cual muestra un ejemplo de cómo se visualiza los datos en la vista materializada.

	NRO_REQ	COD_PROYECTO	COD_CLIENTE	COD_PRIORIDAD	NOMBRE_MODULO	FECHA_INICIO	FECHA_FINAL	NRO_FORMAS	NRO_SP	NRO_REPORTES	NRO_TABLAS	TIE
1	137945	BANESCO-ADP-SOP	BANESCO	BAJA	Tesorería / Inversiones	03-01-2011	02-02-2011	0	2	0	0	
2	138423	GESTOR-V1-INT	GESTOR	BAJA	Tesorería / Inversiones	11-01-2011	13-01-2011	1	0	1	0	
3	140859	CTH-SOP	CTH	BAJA	Crédito	08-02-2011	08-02-2011	0	1	0	0	
4	141082	FIDUCOL-IMP7	FIDUCOL	BAJA	Tesorería / Inversiones	09-02-2011	09-02-2011	1	0	0	0	
5	142485	FODEVASA-SOP	FODEVASA	BAJA	Crédito	10-03-2011	11-03-2011	1	0	0	0	
6	144582	GESTOR-V1-INT	GESTOR	BAJA	Producto	12-04-2011	14-04-2011	1	0	1	0	
7	146220	FODEVASA-SOP	FODEVASA	BAJA	Caja / Bancos	09-05-2011	09-05-2011	1	0	0	0	
8	146440	GESTOR-V1-INT	GESTOR	BAJA	Personas	20-05-2011	31-05-2011	0	0	1	0	
9	146630	GESTOR-V1-INT	GESTOR	BAJA	Producto	27-03-2012	27-03-2012	1	0	0	0	
10	144919	FIDUCOL-SOP	FIDUCOL	BAJA	Caja / Bancos	19-04-2011	23-05-2011	0	0	0	0	
11	147279	GESTOR-V1-INT	GESTOR	BAJA	Herramienta	08-06-2011	13-06-2011	4	0	0	0	
12	146940	BANESCO-TT-SOP	BANESCO	BAJA	Transacciones	02-06-2011	28-07-2011	0	2	0	0	
13	149533	GESTOR-V1-INT	GESTOR	BAJA	Producto	18-07-2011	21-07-2011	1	0	0	0	
14	149721	GESTOR-V1-INT	GESTOR	BAJA	Caja / Bancos	15-07-2011	18-07-2011	2	0	0	0	
15	149020	FIDUCOL-IMP6	FIDUCOL	MEDIA	Herramienta	28-06-2011	22-07-2011	8	1	0	0	

Figura 12 Información de Vista Materializada

Para la ejecución de esta tarea se utiliza una vista materializada, la cual mejora los tiempos de acceso (lectura de registros); adicionalmente la misma puede ser

modificada de acuerdo a las necesidades empleando solo conocimientos del lenguaje SQL y sin recurrir a mucho esfuerzo; optimizando así el tiempo de la investigación. Otra forma a la que pueden recurrir algunos investigadores es a utilizar filtros y facilidades de herramientas ETL o de minería de datos; sin embargo, debido a las posibles limitaciones de memoria o cantidad de registros para proceso que pueden traer estas herramientas, se recomienda realizar todas estas tareas utilizando SQL.

Verificación la calidad de los datos

El tiempo invertido en esta tarea; fue mínimo tomando en cuenta los aspectos que se detallan a continuación:

- El cambio en los procesos de desarrollo de software de la empresa y estabilización de los mismos; hecho por el cual se ha tomado información solo desde el año 2010 en adelante.
- La constante actualización y corrección de la aplicación ISOv2; desde el año 2006 hasta el año 2010; que fue incorporando medidas para garantizar el correcto registro de información por parte de los colaboradores de la empresa; evitando así el registro de información incompleta inválida.
- El uso de la información contenida en la base de datos del sistema ISOv2 como base para la gestión de los procesos financieros, técnicos y de operación; por lo que la calidad de la misma ha sido sometida a verificaciones de forma constante.

Al final una vez que se ha descartado la información de los años pasados; se cuenta con 19410 registros; los cuales serán analizados para su uso previo a la construcción del modelo de minería de datos en la fase 4 de la metodología.

4.2.3 Fase 3 - Preparación de los datos

Selección de Datos

El conjunto de registro obtenidos en la fase 2, cuenta con información de todos los productos de Gestor; al ser cada producto desarrollado en una diferente arquitectura de software; se ha decidido descartar a todos los registros de requerimientos pertenecientes a productos como *"Gestor G5 Trust"* y *"Gestor Banca Electrónica"*.

De acuerdo a lo revisado en la sustentación teórica de este proyecto se ha tomado en cuenta la recomendación de varios autores, en las que se asegura un mejor índice de precisión en los resultados obtenidos; cuando la información se refiere a un grupo de proyectos similares. La diferencia radical entre la plataforma de desarrollo *"Oracle Forms & Reports"* y la plataforma *"Java Enterprise Edition (JEE)"*; ha motivado la decisión de seleccionar como información válida para la construcción del modelo sola la referente al producto *"Gestor Web Fiducia Fondos"*. Esta decisión motivo la remoción de 6000 registros aproximadamente; quedando al final 13524 registros disponibles para las siguientes actividades.

Limpieza de Datos

En esta tarea, se procede a eliminar los registros que no aportan información al modelo que se va a definir. Una vez que se ha revisado la información se ha determinado las siguientes operaciones de limpieza a ser ejecutadas.

1. Eliminar todos los registros cuyo campo: NUM_OBJETOS sea igual a 0. El campo NUM_OBJETOS, contiene la sumatoria de objetos del sistema creados o modificados. Los objetos del sistema se clasifican en 4 categorías; y corresponde a los parámetros que miden el tamaño de un requisito. Los tipos de objetos válidos para esta sumatoria son: número de tablas, número de reportes, número de funciones o procedimientos almacenados y número de pantallas.
2. Eliminar todos los registros que correspondan a tareas de soporte o consulta.

3. Eliminar registros correspondientes a tareas de elaboración de manuales de usuario.
4. Eliminación de valores atípicos. Se consideró valores atípicos a registros cuyo valor de tiempo real era mayor a 150 horas. Que es el estimado de trabajo de una persona al mes.

Una vez que se ha ejecutado estas operaciones, de limpieza el conjunto de datos se ha reducido en un 67% de su tamaño original aproximadamente; dejando 4324 registros para la ejecución de las siguientes fases. Algo interesante que se notó al analizar la información, es que con las operaciones de limpieza; se obtuvo registros válidos sólo desde al año 2011; la razón es que recién desde ese año se empezó a utilizar para la estimación del esfuerzo una métrica de tamaño dada por el número de tablas, objetos de programación, pantallas y reportes.

Estructuración, Integración y Formateo de los datos

Para cumplir con esta tarea se ha exportado solo la información que puede ser utilizada por los algoritmos de minería de datos; es decir, se ha descartado columnas generadas inicialmente en la Fase 2; a continuación se describe la estructura final del archivo que se utilizará para aplicar la técnica de minería de datos (Redes Neuronales) y generar el modelo.

Tabla 10
Estructura del Archivo para aplicación de Técnica de Minería de Datos

Campo	Descripción
COD_DIFICULTAD	Parámetro que indica la dificultad del desarrollo del requerimiento evaluada por el experto en el sistema. Toma los siguientes valores: 1. Baja, 2. Media, 3. Alta, 4. Muy Alta
NRO_FORMAS	Especifica el número de pantallas que se debe crear o modificar para cumplir con la funcionalidad del requerimiento
NRO_OBJ_BD	Especifica el número de: tablas, paquetes, procedimientos almacenados, funciones o disparadores; que deben ser desarrollados o modificados.
NRO_REPORTES	Especifica el número de reportes que se necesita modificar o desarrollar como parte del requerimiento.
TIEMPO_FINAL	Tiempo final real que tomó el desarrollo del requerimiento. Este tiempo incluye: tiempo de desarrollo, tiempo de pruebas y tiempo de re-proceso. El tiempo esta expresado en horas.

4.2.4 Fase 4 - Modelado

Selección de Técnica de Minería de Datos

El algoritmo de minería de datos que se ha seleccionado para la creación del modelo en base a la información disponible es una red neuronal artificial.

La red neuronal es de tipo "*Perceptrón*" y utiliza una capa oculta; vale destacar que contrario a lo que se puede suponer el uso de más de una capa oculta muchas veces

puede ser contraproducente. El algoritmo de aprendizaje que utiliza la red es retro propagación (back propagation).

Para facilitar la implementación del modelo se utiliza la herramienta "*RapidMiner*[4]", la cual cuenta con la implementación del algoritmo de redes neuronales de acuerdo a los requerimientos que se ha definido. Esta herramienta en su versión para la comunidad (libre de costo) permite trabajar hasta con diez mil registros, por lo que no presenta ningún impedimento de uso.

Generación de Plan de Pruebas

Se ha decidido junto con el personal de la empresa utilizar la información de los años 2011 al 2015, que constituye aproximadamente el 83% de la totalidad de los datos disponibles para realizar la tarea de "entrenamiento" de la red neuronal artificial. Para la validación del modelo obtenido se utilizará los datos correspondientes al año 2016.

Construcción del Modelo

De acuerdo a la información que tiene registrada la empresa Gestor referente al esfuerzo real que toma el desarrollo de un requerimiento se puede determinar las siguientes variables:

- **Variable Dependiente**: Tiempo Real que toma el desarrollo de un requerimiento.
- **Variables Independientes**: Dificultad, número de pantallas, número de objetos de base de datos, número de reportes.

La red cuenta con 4 parámetros de entrada, en este caso las variables independientes y una neurona de salida, variable dependiente que corresponde al tiempo estimado de esfuerzo para cada requerimiento. La estimación total del tiempo del proyecto esta dada por la suma del tiempo estimado de cada requerimiento.

[4] RapidMiner es un programa informático para el análisis y minería de datos. Permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico. https://rapidminer.com/

En base a lo expresado anteriormente la ***Figura. 13***, muestra la red neuronal artificial (RNA) construida y que será la encargada de predecir el tiempo de estimación de un requerimiento en base a las variables señaladas.

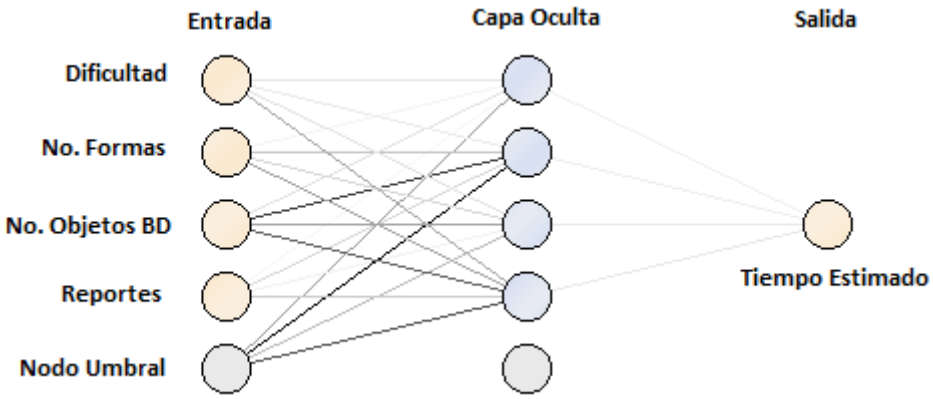

Figura. 13 Modelo de RNA construido por RapidMiner

La ***Figura. 14***, muestra el proceso de minería de datos construido con la herramienta RapidMiner, el proceso esta compuesto por la extracción de los datos de entrenamiento los cuales han sido aislados en una entidad de base de datos, creada especificamente para este proceso; esto también se realizó para los datos de validación. Finalmente como se puede observar los resultados son almacenados en una entidad diferente para su posterior anális y validación.

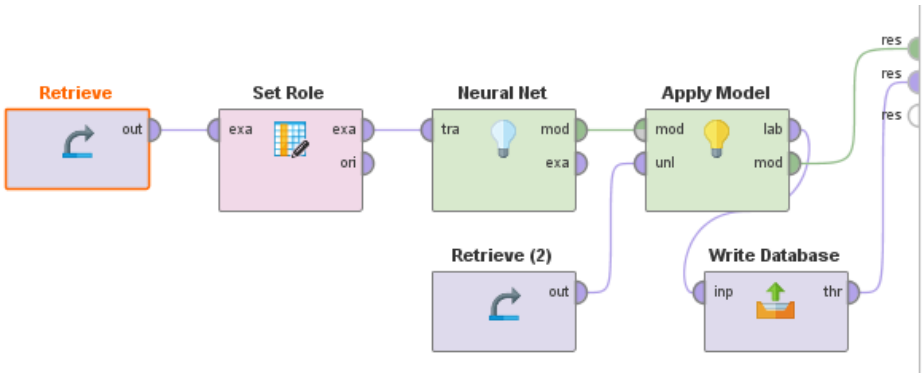

Figura. 14 Proceso creado en la herramienta RapidMiner

Validación del Modelo

Para realizar la validación del modelo cosntruido se ha procedido a revisar los principales nodos que forman parte del proceso de RapidMiner, asegurando la validez

de los parámetros de entrada y salida de cada uno. A continuación se procede a detallar la revisión realizada sobre estos.

1. Nodo Retrieve: este nodo es el encargado de leer los datos que van a ser utilizados para el entrenamiento de la red neuronal artificial. La ***Figura. 15*** , muestra las columnas y la información asociada a las mismas.

Row No.	DIFICULTAD (integer) *regular*	FORMAS (integer) *regular*	OBJETOS_DB (integer) *regular*	REPORTES (integer) *regular*	TIEMPO_TOTAL (real) *regular*
1	5	1	1	0	29.400
2	4	0	2	0	19.900
3	4	0	1	1	13.990
4	2	3	3	0	9.690
5	3	1	1	0	6.400
6	4	1	1	0	16.050
7	5	0	4	0	80.120
8	3	1	1	0	7.740
9	3	0	1	1	9.060

Repository Location: //Local Repository/data/TRAIN1_TESIS

Figura. 15 Muestra de los datos de entrenamiento.

2. Nodo NeuralNet: este nodo representa a la red neuronal artificial que la herramienta RapidMiner construye en base a la información de entrada. El principal parámetro a configurar es el *"número de ciclos de aprendizaje"*, por defecto la herramienta fija este valor a 500 ciclos; se debe tener en cuenta que no siempre a mayor número de ciclos de aprendizaje el resutlado será mejor.
3. Nodo Retrieve(2): este nodo extrae de la base de datos la información que será utilizada para la predicción; en este caso todos los registros cuya fecha en mayor al 1 de enero del 2016. La estructura es similar a la de los datos de entrenamiento.
4. Nodo Write Database: este nodo se encarga de registrar los valores de de predicción en un tabla de la base de datos, para facilitar el posterior análisis de los mismos. La estructura de la salida, será revisada en el siguiente capítulo.

CAPITULO V
DISCUSIÓN DE RESULTADOS

5.1 Introducción

Para el desarrollo de este capítulo se ha decidido utilizar las fases 5 de la metodología CRISP-DM. La fase 5, Evaluación; permitirá realizar una evaluación de los resultados obtenidos con el modelo creado en la fase anterior en base a los "criterios de éxito" definidos en la fase 1 de la metodología.

5.2 Evaluación de los Resultados Obtenidos

El proceso de evaluación de los resultados obtenidos fue un proceso iterativo, en el cual se realizó tres iteraciones diferentes, en las que se realizó ajustes a los parámetros del operador NeuralNet para tratar de disminuir el error encontrado en los resultados de estimación. Adicionalmente aprovechando las ventajas que brinda la herramienta RapidMiner, se construyo un proceso de validación, el cuál permite obtener los valores promedio de error absoluto, error relativo y raiz del error cuadrático de forma automática. A continuación se explica la construcción del proceso de evaluación y se detalla los resultados obtenidos en cada una de las tres iteraciones realizadas.

5.2.1 Diseño del Proceso de Evaluación

Para la construcción del proceso de evaluación, se utilizó un nodo especial de la herramienta RapidMiner llamado *"Cross Validation"* (Validación Cruzada)[5]; este nodo es un agrupador de procesos, que permite definir internamente el modelo a ser evaluado, y realizar la evaluación de rendimiento del mismo. La ***Figura. 16***, muestra el diseño general del proceso de evaluación; mientras que la ***Figura. 17***, muestra la configuración interna del nodo *"Cross Validation"*.

[5] La validación cruzada es un método estadístico estándar para estimar la generalización del error en un modelo predictivo. https://jeszysblog.wordpress.com/2012/04/13/cross-validation-in-rapidminer/

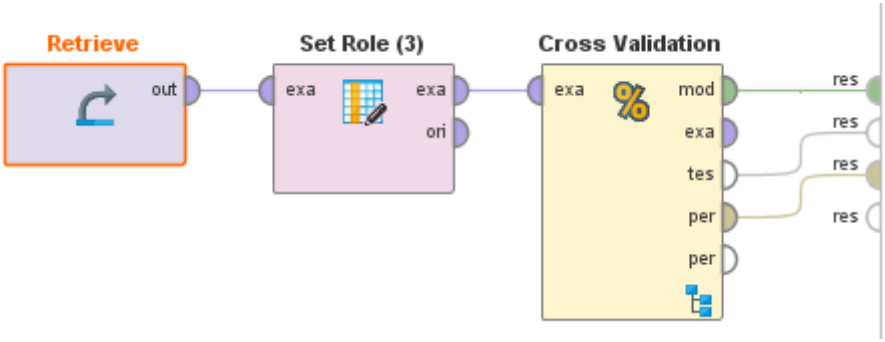

Figura. 16 Diseño general del proceso de evaluación.

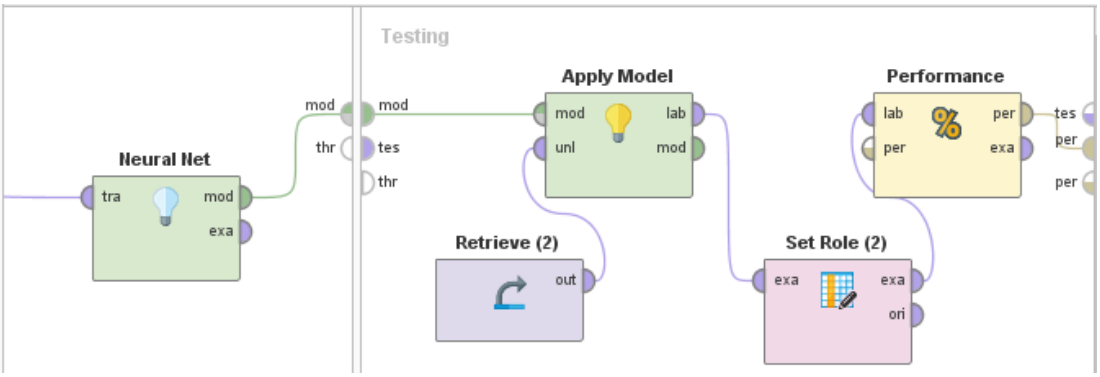

Figura. 17 Configuración interna del nodo “Cross Validation”

Como se muestra en la ***Figura. 17***, el nodo “*Performance”* es el que se encarga de realizar los cálculos del promedio de error absoluto, error relativo y raiz del error cuadrático de forma automática; una vez que el modelo ha sido entrenado y se ha realizado la predicción para los datos del conjunto de prueba.

5.2.2 Análisis de Resultados

Como se explicó anteriormente se realizaron tres iteraciones diferentes en cada una de las cuales se modificó el número de ciclos de entrenamiento de la red neuronal, para tratar de disminuir los valores de error calculados. A continuación se describe cada una de estas iteraciones.

Iteración 1

Para esta iteración se definió un valor de 250 ciclos de aprendizaje, obteniendo los siguientes resultados de rendimiento:

Tabla 11
Valores de error para 250 ciclos de aprendizaje.

Error Absoluto	6.265 +/- 1.756
Error Relativo	95.68% +/- 52.71%
Raiz Error Cuadratico Medio	10.271 +/- 1.746
Promedio de Predicción	16.006 %

Iteración 2

Para esta iteración se definió un valor de 500 ciclos de aprendizaje (valor por defecto de la herramienta RapidMiner), obteniendo los siguientes resultados de rendimiento:

Tabla 12
Valores de error para 500 ciclos de aprendizaje

Error Absoluto	6.243 +/- 1.816
Error Relativo	94.97% +/- 56.73%
Raiz Error Cuadratico Medio	10.170 +/- 1.760
Promedio de Predicción	16.006 %

Iteración 3

Para esta iteración se definió un valor de 900 ciclos de aprendizaje, obteniendo los siguientes resultados de rendimiento:

Tabla 13
Valores de error para 900 ciclos de aprendizaje

Error Absoluto	6.083 +/- 1.753
Error Relativo	92.66% +/- 54.36%
Raiz Error Cuadratico Medio	10.024 +/- 1.745
Promedio de Predicción	16.006 %

Como se puede apreciar en los resultados obtenidos; el número de ciclos de aprendizaje de la red neuronal artificial, mejora los valores de los errores obtenidos; sin embargo, los mimos son extremadamente altos.

El valor del promedio de predicción para los 3 casos fue el mismo; lo que nos indica que solo el 16% de los casos analizados fueron correctos; lo cual no es una señal alentadora para la implantación del mismo.

Al obtener estos valores; se realizó varias iteraciones volviendo a las fases 2 y 3 de la metodología; para tratar de encontrar datos que podrían afectar al modelo, sin embargo, no se encontraron casos relevantes ya que al ser descartados no mejoraron el rendimiento del modelo planteado.

Analizando un poco más a detalle y para tratar de entender donde se encuentran las fallas del modelo de estimación; se agregó manualmente una columna en la tabla de resultado, para calcular el error relativo de cada registro y proceder a realizar agrupaciones que nos permitan entender de una mejor forma los valores de error altos encontrados en el modelo.

Una vez que se ha calculado el error relativo de cada registro, se realiza una agrupación en base a rangos del valor del mismo obteniendo los siguientes resultados.

Tabla 14
Registros en base a rangos de error relativo

Rango Error Relativo	Registros	Porcentaje
0%-10%	173	24%
10%-25%	186	26%
25%-50%	153	21%
50%-100%	108	15%
Mayores 100%	100	14%

De acuerdo a esta tabla; se observa que el 50% del total de registros presenta un error relativo de estimación menor al 25%. Para entender de mejor forma los registros

cuyo error relativo de estimación es mayor al 25%; se procedío a agruparlos en base al total del número de objeto de base de datos, formas y reportes obteniendo la siguiente información.

Tabla 15
Casos de error en base al total de objetos

Total objetos	Requerimientos con Error Relativo mayor al 25%	Porcentaje
2	199	55.1%
4	56	15.5%
6	25	6.9%
8	11	3.6%
10	13	3.0%
Resto	57	15.8%

De acuerdo a esta clasificación se puede observar que los registros que presentan un mayor error relativo corresponden a requerimientos que tienen de 2 a 6 objetos de desarrollo es decir requerimientos pequeños; los cuales sumados dan el 77% de los casos.

En una revisión de los registros cuyo total de objetos es menor o igual a 6 y que tienen un error relativo mayor al 25%; se pudo observar que en el 60% de los casos corresponden a requerimientos que duraron hasta 4 horas, como se muestra en la siguiente tabla.

Tabla 16
Registros con mayor error en base al tiempo de desarrollo

Tiempo Desarrollo	Casos	Porcentaje
Menor a 2 horas	82	29.3%
De 2 a 4 horas	85	30.4%
De 4 a 8 horas	41	14.6%
De 8 a 16 horas	43	15.4%
Mayores a 16 horas	29	10.4%

De acuerdo a todas las revisiones realizadas anteriormente se puede deducir que la falla del modelo se da principalmente en los requerimientos pequeños cuyo tiempo de desarrollo tomo hasta un día de trabajo.

Los requerimientos pequeños (de hasta 6 objetos y menores a 1 día de trabajo) constituyen la mayoría de los casos utilizados en el conjunto de entrenamiento y en el de prueba; la falla en la estimación de los tiempos de estos registros si bien puede tener un error relativo alto; al ser menores a un día de trabajo; no cosntituirían una falla tan grande al calular el total del tiempo de un proyecto.

Se presume, que las fallas de estimación estan dadas en que hay demasiados registros similares, es decir con el mismo valor de dificultad, número de formas, número de objetos de bd y número de reportes con tiempos de desarrollo que va de 0.5 a 8 horas; por lo que al ser esta la principal razón de la inexactitud del modelo, la toma de acciones correctivas se vuelve más compleja.

Comparación de Resultados con Juicio Experto

Se ha tomado el tiempo estimado para cada requerimientos dado por los líderes de proyecto de la empresa Gestor, para comparar su fiabilidad versus la fiabilidad del modelo generado y analizado anteriormente; obteniendo así los siguientes resultados.

Tabla 17
Valores de Error para Juicio Experto

Error Absoluto	4.952
Error Relativo	53.22%
Raiz Error Cuadratico Medio	7.29

De acuerdo a estos resultados; se puede concluir que la fiabilidad del Juicio Experto es mayor a la del modelo generado. Analizando al igual que en el modelo generado, se ha procedido a agrupar los casos de error obteniendo la siguiente información:

Tabla 18
Registros en base a rangos de error relativo Juicio Experto

Total objetos	Requerimientos con Error Relativo mayor al 25%	Porcentaje
2	177	25%
4	183	25.4%
6	177	25%
8	75	9.6%
10	32	4%
Resto	74	10%

5.2.3 Proceso de Revisión

Una vez que se ha dado por finalizado el proceso de modelamiento y su evaluación, es necesario revisar todo el proceso de minería de datos; en este caso al encontrar valores de error que sobrepasaban las espectativas tanto del investigador como de los ejecutivos de la empresa, se realizaron varias iteraciones tratando de encontrar posibles fallas; sin embargo, estas no fueron detectadas; las operaciones de selección, validación, formateo y liempieza de los datos se consideraron válidas.

El análisis detallado de los valores de error, sirvió para llegar a la causa de falla del modelo; estas fallas nos han llevado a concluir que la información que actualemente la empresa esta registrando de sus desarrollos no es suficiente para poder generar un modelo de minería de datos que ayude y mejore el porcentaje de fiabilidad que actualmente tiene el método de Jucio Experto que es utilizado por la empresa.

5.2.4 Determinación de Futuras Fases

En base a los resultados obtenidos; y a que la fiabilidad del Juicio Experto actualmente utilizado por la empresa Gestor es mejor que el del modelo generado; se ha decicido no implementar el modelo.

Gracias al análisis realizado, se ha detecctado que se debe mejorar la recolección de la información de los proyectos de la empresa; para que la misma pueda ser utilizada en el futuro para estimaciones que produzcan mayor beneficios a la empres; adicionalmente este proyecto servirá como base para la creación de nuevos proyectos que ayuden a la empresa a entender su información.

CAPITULO VI
CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

En este caso de estudio se ha contado con toda la información real necesaria para el desarrollo de la investigación; es la primera vez que la empresa ha utilizado la información almacenada para predecir el futuro y aunque el resultado no haya sido el esperado ha sido una gran experiencia para todos los involucrados. Una vez que el proceso de investigación ha sido conluido se ha llegado a las siguientes conclusiones:

- En primer lugar, se puede concluir que en base a la información que posee la empresa Gestor de sus proyectos; no se ha podido construir un modelo de predicción que mejore el porcentaje de fiabilidad en la estimación en comparación a los resultados obtenidos por el método de Juicio Experto, actualmente utilizado.

- En segundo lugar; se llego a determinar que la mayor cantidad de casos fallidos de estimación se encuentra en requerimientos pequeños; es decir requerimientos de menos de un día de desarrollo; gracias a esta conclusión se ha determinado la necesidad de registrar más información de los proyectos; la cual se convierta en nuevas entradas de un modelo de redes neuronales aritificiales que permita mejor su precisión.

- Finalmente; gracias al proceso de minería de datos, se descubrió que muchos registros presentaban inconsistencias en base a lo que se debía realizar y al tiempo registrado por los desarrolladores; es decir, se llego a detectar que muchos desarrolladores registraban más tiempo del que realmente debían haber empleado.

6.2. Trabajo Futuro

La precisión de un modelo de estimación como el que se desarrolló en esta investigación depende de las variables independientes seleccionadas y la calidad de información de las mismas; por este motivo la empresa ha tomado dos decisiones principales para el futuro.

La primera consiste en registrar además del número de objetos de base de datos, formas o reportes a ser modificados o creados; a distinguir claramente el número de objetos creados y el número de objetos a ser modificados; además se ha determinado la necesidad de especificar si los cambios corresponden a cambios masivos o no; con esta nueva información recolectada durante seis meses se procederá a crear un nuevo modelo que incluya estas nuevas variables esperando la mejora de la fiabilidad del modelo.

La segunda decisión está basada en el elemento humano; es necesario concientizar a los desarrolladores que el tiempo que ellos registran es de vital importancia para la empresa; por lo cual el mismo no debería ser “ajustado” a sus intereses.

Una vez que se ha emprendido el primer proyecto de minería de datos, se ha visto una posibilidad de generar proyectos destinados al análisis de la información de los cleintes de la empresa; para venderlos como nuevos proyectos de consultoría.

BIBLIOGRAFIA

AESOFT. (2016). *3er Benchmark del Sector de la Industria del Software.* Quito.

Almache, M. G., Raura, G., Ruiz, J. A., & Fonseca, E. R. (2015). Modelo neuronal de estimación para el esfuerzo de desarrollo en proyectos de software (MONEPS). *Revista Latinoamericana de Ingenieria de Software, 3*(3), 148-154.

Aluja, T. (2001). La minería de datos, entre la estadística y la inteligencia artificial. *QÜESTIIO, 25*(3), 479-498.

Arancibia, J. (2009). *Metodología para la definición de requisitos en proyectos de data mining (er-dm).* Madrid: Universidad Politécnica de Madrid.

Bourque, P., & Fairley, R. E. (2014). *SWEBOK: Guide to the Software Engineering Body of Knowledge v3.0.* Los Alamitos, CA: IEEE Computer Society.

Calderón, A., Castillo, M., & Bercovich, N. (2013). *La cadena del software en Ecuador: Diagnóstico, visión estratégica y lineamientos de política.* Quito: CEPAL.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & & Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step data mining guide.*

Curtis, B., Hefley, W., & Miller, S. (2001). *People Capability Maturity Model.* Addison-Wesley.

Fairley, R. (2009). *Managing and Leading Software Projects.* Los Alamitos, CA: Wiley-IEEE Computer Society.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Advances in Knowledgeand Data Mining, 39*(11).

Forrester, J. (1960). *Industrial Dynamics.* Pegasus Communications.

García, A., González, I., Colomo, R., López, J., & Ruiz, B. (2011). Metodología para la optimización de la estimación del desarrollo de software utilizando redes neuronales. *IEEE Latin America Transactions*, 391-405.

García, J., & Molina, J. (2012). *Técnicas de Análisis de Datos.* Madrid.

González, A. (30 de Julio de 2014). *CleverData.io.* Recuperado el 5 de Enero de 2017, de http://cleverdata.io

Graupe, D. (2007). *PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS (2nd Ed.).* Singapore: World Scientific.

Humboldt State University. (3 de Marzo de 2015). *Humboldt State University.* Recuperado el 15 de Marzo de 2017, de https://www2.humboldt.edu/its/sites/default/files/docs/po-docs/Software%20Selection%20Process%20March%202015%20final.pdf

Izquierdo, L., Ordax, J., Santos, J., & Martínez, R. (2008). Modelado de sistemas complejos mediante simulación basada en agentes y mediante dinámica de sistemas. *Empiria. Revista de metodología de ciencias sociales*(16), 85-112.

Jones C. (2007). *Estimación de costos y Administración de proyectos de Software (2 ed.).* México: McGraw-Hill.

Jørgensen, M. (2004). A review of studies on expert estimation of software development effort. *Journal of Systems and Software, 70*(1), 37-60.

Kenny, P. (2014). *Better Business Decisions from Data: Statistical Analysis for Professional Success.* APress.

Kirk, M. (2015). *Thoughtful Machine Learning.* Sebastopol, CA: O'Reilly Media.

López, J., & Dolado, J. (2007). Estudio de los métodos de estimación: AHP y redes Bayesianas. *Departamento de Lenguajes y Sistemas, Universidad del País Vasco*.

Luna, E., Álvarez, F., Espinaoza, M., Ambriz, H., & Nungaray, A. (2010). Modelo para Almacenar y Recuperar Métricas de Software. *Conciencia Tecnológica*(39), 31-37.

Melany Gualavisí M., S. M. (2011). El Sector de la Cibernética: una primera aproximación. *Boletín mensual de análisis sectorial de MIPYMES - Sector de la Cibernética*, 16-17.

Moine, J., Haedo, A., & Gordillo, S. (2011). Estudio comparativo de metodologías para minería de datos. Obtenido de http://sedici.unlp.edu.ar/bitstream/handle/10915/20034/Documento_completo.pdf?sequence=1

Moro, S., Laureano, R., & Cortez, P. (2011). Using data mining for bank direct marketing: An application of the crisp-dm methodology. *Proceedings of European Simulation and Modelling Conference-ESM*, 117-121.

Pérez, A. L., González, L., Duque, A., Millane, F., & Ospina, G. (2006). Modelo dinámico para la estimación temprana de esfuerzo en proyectos de desarrollo de software. *Revista Ingenierías Universidad de Medellín*, 11-20.

Pérez, C. (2004). *Técnicas de análisis multivariante de datos.* Madrid: Pearson.

Pérez, M. (2014). *Minería de Datos a través de Ejemplos.* Madrid: RC Libros.

Piatetsky, G. (Octubre de 2014). *CRISP-DM, still the top methodology for analytics, data mining, or data science projects - KDnuggets TM.* Recuperado el 5 de Marzo de 2017, de KDnuggets TM: http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

Pressman, R. S. (2010). *Software engineering: A practitioner's approach (7th ed.).* Boston, Mass.: McGraw Hill.

Project Management Institute. (2004). *A guide to the project management body of knowledge (PMBOK guide 5th ed.).* Newtown Square, Pa: Project Management Institute.

Quintero, L., Antón, J., Ferreira, G., & Gálvez, D. (2013). Aprendizaje Automático en la Estimación de Tiempo en Tareas de Proyectos de Software. *II Conferencia Internacional de Ciencias Computacionales e Informáticas*.

Robiolo, G., Castillo, O., Rossi, B., & Santos, S. (2013). Es posible superar la precisión basada en el juicio de expertos de la estimación de esfuerzo de productos de software? *X Workshop Latinoamericano Ingeniería de Software Experimental*.

Salvetto, P., Nogueira, J. C., & Segovia, J. (2004). Modelos Automatizables de Estimación muy Temprana del Tiempo y Esfuerzo de Desarrollo de Software de Gestión. *XXX Conferencia Latinoamericana de Informatica, 129*.

Scott, G. (2001). Strategic planning for high-tech product development. *Technology Analysis & Strategic Management, 13*(3), 343-364.

Sommerville, I. (2011). *Software Engineering (9th ed.).* Boston, MA: Pearson Addison-Wesley.

Subsecretaría de Gestión Estratégica e Innovación. (2011). *Manual de Contratación Pública.* Quito: Secretaría Nacional de la Administración Pública.

Timarán, S., Hernández, I., Caicedo, S., Hidalgo, A., & Alvarado, J. (2016). El proceso de descubrimiento de conocimiento en bases de datos. *Descubrimiento de patrones de desempeño académico con árboles de decisión en las competencias genéricas de la formación profesional*, 63-86.

Trendowicz, A., & Jeffery, R. (2014). *Software Project Effort Estimation.* Switzerland: Springer.

Trendowicz, A., Münch, J., & Jeffery, R. (2008). State of the practice in software effort estimation: a survey and literature review. *FIP Central and East European Conference on Software Engineering Techniques*, 232-245.

Varas, M. (2002). *Una Experiencia con la estimación del tamaño del software.* Recuperado el 25 de 02 de 2017, de http://www.inf.udec.cl/revista/edicion1/mvaras.htm.

Vela Casado, C. (2010). *La industria de software: una experiencia de empresas, gobiernos y universidades en Uruguay y Ecuador.* Quito, Ecuador: Flacso Ecuador.

Velthuis, M., Villalón, J., Bravo, J., & Sanz, L. (2003). *Análisis y diseño de aplicaciones informáticas de gestión: una perspectiva de ingeniería del software.* Ra-ma.

Witten, I., Frank, E., & Hall, M. (2011). *Data Mining - Practical Machine Learning Tools and Techniques (3th ed.).* Burlington, MA: Morgan Kaufmann.