



UNIVERSIDAD CATÓLICA ANDRÉS BELLO  
VICERRECTORADO ACADEMICO  
DIRECCIÓN GENERAL DE LOS ESTUDIOS DE POSTGRADO  
AREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN  
POSTGRADO EN GERENCIA DE PROYECTOS

TRABAJO ESPECIAL DE GRADO

**ANALISIS DE LA ESTIMACIÓN DE ESFUERZO EN PROYECTOS DE SOFTWARE  
UTILIZANDO PUNTO DE FUNCIÓN Y REDES NEURONALES**

Presentado por  
Martínez Martineau Roger Antonio

Para optar al título de  
Especialista en Gerencia de Proyectos

Asesor  
Ing. Mauricio Paletta Msc.

Ciudad Guayana, Noviembre de 2008

UNIVERSIDAD CATÓLICA ANDRÉS BELLO  
VICERRECTORADO ACADEMICO  
DIRECCIÓN GENERAL DE LOS ESTUDIOS DE POSTGRADO  
AREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN  
POSTGRADO EN GERENCIA DE PROYECTOS

TRABAJO ESPECIAL DE GRADO

**ANÁLISIS DE LA ESTIMACIÓN DE ESFUERZO EN PROYECTOS DE SOFTWARE  
UTILIZANDO PUNTO DE FUNCIÓN Y REDES NEURONALES**

Presentado por  
Martínez Martineau Roger Antonio

Para optar al título de  
Especialista en Gerencia de Proyectos

Asesor  
Ing. Mauricio Paletta Msc.

Ciudad Guayana, Noviembre de 2008

UNIVERSIDAD CATÓLICA ANDRÉS BELLO  
VICERRECTORADO ACADÉMICO  
ESTUDIOS DE POSTGRADO  
ÁREA DE GERENCIA  
POSTGRADO EN GERENCIA DE PROYECTOS

**ACEPTACION DEL ASESOR**

Por la presente hago constar que he leído el Trabajo Especial de Grado, presentado por el ciudadano **Roger Antonio Martínez Martineau**, para optar al grado de **Especialista en Gerencia de Proyectos**, cuyo título es “**Análisis de la Estimación de Esfuerzo en Proyectos de Software utilizando Punto de Función y Redes Neuronales**”; y manifiesto que cumple con los requisitos exigidos por la Dirección de los Estudios de Postgrado de la Universidad Católica Andrés Bello, por lo tanto, lo considero apto para ser evaluado por el jurado que se decida designar a tal fin.

En la ciudad de Puerto Ordaz, a los doce días del mes de Septiembre de 2008.

---

Ing. Mauricio Paletta Msc.

C.I: 6.560.965

## **DEDICATORIA**

Quiero dedicar este trabajo de investigación a mi padre, quien me enseñó con sus acciones a luchar en la vida bajo condiciones adversas, a ser paciente y nunca rendirse cuando se tiene un objetivo, y enseñarme que el amor es un sentimiento tan fuerte que supera nuestro propio entendimiento.

Así mismo, quiero dedicar este trabajo a mi esposa, por estar presente en los momentos más difíciles que hasta ahora he afrontado, a mi madre por enseñarme el valor del trabajo y a mi hermana por inculcarme el deseo de pensar en grande.

## **AGRADECIMIENTOS**

Quiero agradecer a todos mis compañeros de clases, por hacer de esta especialización, un paseo lleno de momentos agradables e inolvidables. A mi esposa Jackeline por apoyarme en todo momento y a mi antiguo profesor de pregrado y actual tutor, Ingeniero Mauricio Paletta, por ayudarme en mi crecimiento profesional desde etapas muy tempranas hasta la actualidad.

## ÍNDICE GENERAL

	Pág
ACEPTACION DEL ASESOR.....	ii
DEDICATORIA .....	iii
AGRADECIMIENTOS .....	iv
ÍNDICE DE TABLAS.....	vii
ÍNDICE DE FIGURAS.....	viii
ÍNDICE DE GRÁFICAS.....	ix
RESUMEN.....	xii
INTRODUCCION.....	1
<b>CAPÍTULO I. PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS DE LA INVESTIGACIÓN .....</b>	<b>2</b>
<b>1. PLANTEAMIENTO DEL PROBLEMA.....</b>	<b>2</b>
<b>2. OBJETIVOS DE LA INVESTIGACION .....</b>	<b>10</b>
2.1.    Objetivo General .....	10
2.2.    Objetivos Específicos .....	11
<b>CAPÍTULO II. MARCO CONCEPTUAL .....</b>	<b>12</b>
<b>1. PROYECTOS DE SOFTWARE.....</b>	<b>12</b>
<b>2. ESTIMACIÓN DE PROYECTOS DE SOFTWARE.....</b>	<b>13</b>
<b>3. MÉTODOS DE ESTIMACIÓN DE PROYECTOS DE SOFTWARE.....</b>	<b>17</b>
3.1.    Métodos Algorítmicos .....	19
3.2.    Métodos no Algorítmicos .....	26
<b>4. BASE DE DATOS HISTÓRICOS DE PROYECTOS .....</b>	<b>28</b>
<b>5. INTELIGENCIA ARTIFICIAL (I.A.).....</b>	<b>30</b>
5.1.    Aplicaciones .....	31
5.2.    Técnicas de I.A. aplicadas en la estimación del esfuerzo en el desarrollo de software....	32
<b>6. METODOLOGIA DE SUPERFICIE DE RESPUESTA.....</b>	<b>42</b>
<b>7. TIEMPO DE DESARROLLO Y LA EXPERIENCIA.....</b>	<b>42</b>
<b>CAPÍTULO III. MARCO METODOLÓGICO .....</b>	<b>45</b>
<b>1. TIPO DE INVESTIGACIÓN.....</b>	<b>45</b>

<b>2. UNIDAD DE ANÁLISIS, POBLACIÓN Y MUESTRA .....</b>	<b>45</b>
2.1. Unidad de Análisis .....	45
2.2. Población .....	46
2.3. Tamaño de la Muestra .....	46
<b>3. VARIABLES .....</b>	<b>46</b>
3.1. Variables Independientes.....	46
3.2. Variable Dependiente .....	47
<b>4. ESTRATEGIA DE RECOLECCIÓN Y PROCESAMIENTO Y ANÁLISIS.....</b>	<b>47</b>
4.1. Recolección .....	47
4.2. Procesamiento .....	47
4.3. Análisis.....	49
<b>CAPÍTULO IV. PRESENTACIÓN Y ANÁLISIS DE RESULTADOS .....</b>	<b>51</b>
<b>1. SELECCIÓN DE LA ARQUITECTURA DE LA RNA.....</b>	<b>51</b>
<b>2. RESULTADOS DEL ENTRENAMIENTO .....</b>	<b>54</b>
2.1. Modelo 3.....	54
2.2. Modelo 7.....	55
<b>3. ANÁLISIS DEL ESFUERZO PRONOSTICADO A TRAVÉS DEL MODELO 3 .....</b>	<b>56</b>
3.1. Esfuerzo en función a las Entidades y las Transacciones.....	59
3.2. Esfuerzo en función a la Experiencia del Equipo y la Complejidad del Proyecto .....	63
3.3. Esfuerzo en función a la Experiencia del Gerente y Experiencia del Equipo .....	65
3.1. Esfuerzo en función a las Transacciones y el Ajuste.....	70
<b>4. ANÁLISIS DEL ESFUERZO PRONOSTICADO A TRAVÉS DEL MODELO 7 .....</b>	<b>73</b>
4.1. Esfuerzo en función a las entidades y las transacciones.....	75
4.2. Esfuerzo en función a la experiencia del gerente y del equipo.....	77
<b>5. EJEMPLO PRÁCTICO DE USO .....</b>	<b>79</b>
<b>CONCLUSIONES.....</b>	<b>84</b>
<b>RECOMENDACIONES.....</b>	<b>86</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>87</b>

## ÍNDICE DE TABLAS

N°	Pág.
Tabla 1. Matriz de Clasificación de Estimados de Costos para Procesos Industriales.....	15
Tabla 2. Cálculo de Puntos de Función. ....	22
Tabla 3. Ponderación de complejidad para tipos de objeto. ....	24
Tabla 4. Tasa de productividad por puntos objeto.....	25
Tabla 5. Equivalencia en la terminología estadística y de redes neuronales. ....	39
Tabla 6. Equivalencia entre modelos estadísticos y modelos de red neuronal. ....	39
Tabla 7. Resumen estadístico de la variable diferencia.....	48
Tabla 8. Niveles de Valores de los factores utilizados en el diseño. ....	49
Tabla 9. Análisis de Superficie de cada modelo estudiado. ....	52
Tabla 10. ANOVA para diferencia por modelo. ....	53
Tabla 11. Resumen estadístico de la diferencia.....	53
Tabla 12. Coeficiente de Regresión para Esfuerzo del Modelo 3 .....	58
Tabla 13. Coeficiente de Regresión para Esfuerzo del Modelo 7. ....	74
Tabla 14. Matriz para el cálculo de los puntos de función no ajustados del Proyecto “Casa Modelo” .....	80
Tabla 15. Matriz de Cálculo del Ajuste del Proyecto “Casa Modelo” .....	80

## ÍNDICE DE FIGURAS

N°	. Pág.
Figura 1. Neurona Artificial .....	36
Figura 2. Ejemplo de una RNA .....	38
Figura 3. Esquema de los Métodos Interpretativos Propuestos.....	40
Figura 4. Variables Independientes .....	46

## ÍNDICE DE GRÁFICAS

N°	Pág.
Gráfica 1. Efecto más realista de los costes de desarrollo y valor aportado por el producto del proyecto frente a las funcionalidades. ....	8
Gráfica 2. Horas/Hombre x Unidad Producida .....	9
Gráfica 3. Número de meses para completar el proyecto.....	13
Gráfica 4. Original y Estimado del Modelo 3. ....	55
Gráfica 5. Original y Estimado del Modelo 7. ....	56
Gráfica 6. Efectos Principales para Esfuerzo del Modelo 3.....	57
Gráfica 7. Pareto estandarizado para Esfuerzo del Modelo 3.....	58
Gráfica 8. Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 1, Modelo 3. ....	60
Gráfica 9. Contorno de Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 1, Modelo 3. ....	60
Gráfica 10. Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 2, Modelo 3.....	61
Gráfica 11. Contornos de la Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 2, Modelo 3. ....	61
Gráfica 12. Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 3, Modelo 3.....	62
Gráfica 13. Contornos de la Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 3, Modelo 3. ....	62
Gráfica 14. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Ajuste sobre el Esfuerzo para el Lenguaje 1, Modelo 3. ....	63
Gráfica 15. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Ajuste sobre el Esfuerzo para el Lenguaje 2, Modelo 3. ....	64
Gráfica 16. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Ajuste sobre el Esfuerzo para el Lenguaje 3, Modelo 3. ....	65

Gráfica 17. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 1, Modelo 3.....	65
Gráfica 18. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 1, Modelo 3. ....	66
Gráfica 19. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 2, Modelo 3.....	67
Gráfica 20. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 2, Modelo 3. ....	67
Gráfica 21. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 3, Modelo 3.....	68
Gráfica 22. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 3, Modelo 3. ....	68
Gráfica 23. Contornos de la Superficie de Respuesta de la Experiencia del Gerente y el Lenguaje sobre el Esfuerzo, Modelo 3. ....	69
Gráfica 24. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Lenguaje sobre el Esfuerzo, Modelo 3. ....	69
Gráfica 25. Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 1, Modelo 3.....	70
Gráfica 26. Contornos de la Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 1, Modelo 3.....	70
Gráfica 27. Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 2, Modelo 3.....	71
Gráfica 28. Contornos de la Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 2, Modelo 3.....	71
Gráfica 29. Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 3, Modelo 3.....	72
Gráfica 30. Contornos de la Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 3, Modelo 3.....	72
Gráfica 31. Efectos Principales para Esfuerzo del Modelo 7.....	73
Gráfica 32. Diagrama de Pareto Estandarizado para Esfuerzo del Modelo 7. ....	74

Gráfica 33. Superficie de Respuesta de las Transacciones y Entidades sobre el Esfuerzo para el Lenguaje 1, Modelo 7.....	76
Gráfica 34. Superficie de Respuesta de las Transacciones y Entidades sobre el Esfuerzo para el Lenguaje 2, Modelo 7.....	76
Gráfica 35. Superficie de Respuesta de las Transacciones y Entidades sobre el Esfuerzo para el Lenguaje 3, Modelo 7.....	77
Gráfica 36. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 1, Modelo 7.....	77
Gráfica 37. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 2, Modelo 7.....	78
Gráfica 38. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 3, Modelo 7.....	78

UNIVERSIDAD CATÓLICA ANDRÉS BELLO  
VICERRECTORADO ACADEMICO  
DIRECCIÓN GENERAL DE LOS ESTUDIOS DE POSTGRADO  
AREA DE CIENCIAS ADMINISTRATIVAS Y DE GESTIÓN  
POSTGRADO EN GERENCIA DE PROYECTOS

**Análisis de la Estimación de Esfuerzo en Proyectos de Software utilizando Punto de  
Función y Redes Neuronales.**

Nombre del autor: Roger A. Martínez M.

Nombre del asesor: Mauricio Paletta

Año: 2008

TRABAJO ESPECIAL DE GRADO

**RESUMEN**

Actualmente, en etapas iniciales de definición de un proyecto de tecnología de información, se evidencia una gran debilidad en la estimación del esfuerzo necesario para su desarrollo. Por tal motivo, esfuerzos para la búsqueda de métodos de estimación que sean fáciles, rápidos, confiables y objetivos se han venido desarrollando. El juicio del experto es una de las herramientas de mayor utilización en estos casos donde aún no se tiene completa comprensión del problema; sin embargo, se ve afectada por la tendencia a la subjetividad natural del ser humano, la cual se intenta controlar mediante la técnica Delphi. Una rama de la inteligencia artificial se ha venido desarrollando a fin de sustituir el pensamiento de analogía que un experto pueda aplicar con base en sus conocimientos, para poder realizar la estimación de esfuerzo de un proyecto. Considerando la inestabilidad de los requerimientos de los proyectos de software por motivos tales como la falta de comprensión del problema, del sistema de cómputo o por la misma naturaleza cambiante de las necesidades, el actual estudio tiene como objetivo comprender la relación entre el número de funcionalidades de un software, la experiencia del equipo y el esfuerzo necesario para su desarrollo medido en número de horas/hombres, utilizando las redes neuronales como herramienta de inteligencia artificial. Se espera que el resultado de este análisis contribuya al mejor entendimiento del comportamiento de los proyectos de software a medida que la complejidad y la experiencia aumentan o disminuyen basándose en datos reales.

Metodológicamente constituye un estudio no experimental, explicativo y transversal. La unidad de estudio constituye cada proyecto de la base de datos Desharnais. La población total la constituyen todos los proyectos de software terminados y registrados en bases de datos. Las variables independientes son las transacciones y entidades medidas en puntos de función no ajustados, el lenguaje utilizado y la experiencia del equipo de proyecto y su gerente, y la variable dependiente representa el esfuerzo para desarrollar un proyecto medido en horas/hombres.

**Palabras Claves:**

Proyecto, Proyectos de Software, Formulación de Proyectos, Estimación, Esfuerzo, Puntos de Función, Analogía, Redes Neuronales, Inteligencia Artificial.

## INTRODUCCION

Estimar lo que pueda suceder en un proyecto tiene mayor valor cuando la información se obtiene con tiempo suficiente para tomar una decisión al respecto. Los cambios generan un mayor impacto en los costos y en el tiempo a medida que se hace en etapas más tardías del desarrollo del proyecto; en tal sentido, disponer de la información en etapas tempranas es requerido. Esta necesidad se ve incrementada por el modelo económico mundial que cambia vertiginosamente y exige a las empresas una continua adaptación para mantenerse competitivas.

La gerencia del tiempo, los costos, el alcance y los parámetros de rendimiento son aspectos a los que se orientan los gerentes para predecir el éxito de un proyecto con calidad. Esta temática es muy clara en algunas especialidades como la construcción; sin embargo, en proyectos de tecnología de información sucede lo contrario. Estos proyectos poseen un alto grado de incertidumbre por su naturaleza innovadora, y donde la estimación del tiempo representa un gran reto en el momento en que se decide su emprendimiento.

El presente trabajo, además de hacer un sondeo en los diferentes métodos de estimación del esfuerzo en proyectos de software, hace uso de la técnica de inteligencia artificial conocida como redes neuronales, para el análisis superficie de respuesta de variables como la experiencia del equipo de proyecto y del gerente, medido en años, y el alcance del proyecto medido en puntos de funciones (entidades, transacciones y ajustes de complejidad), con la intención de contribuir en la comprensión de su influencia sobre el esfuerzo. Así mismo, se busca dejar claro un conjunto de herramientas para hacer repetibles, verificables o refutables las conclusiones que surjan como resultado del trabajo.

# **CAPÍTULO I.**

## **PLANTEAMIENTO DEL PROBLEMA Y OBJETIVOS DE LA INVESTIGACIÓN**

### **1. PLANTEAMIENTO DEL PROBLEMA**

A lo largo de su existencia, el hombre ha comprendido la necesidad de unir esfuerzos para lograr objetivos comunes, por lo que se han establecido complejas estructuras para ser más eficientes en el logro de metas, dando origen así a las organizaciones. En este sentido, la ampliación de los mercados modernos con la ayuda de la tecnología, ha creado un nivel de competencia más exigente entre las organizaciones, llegando incluso a surgir la hipercompetencia, que expone la necesidad de generar cambios para lograr mayor velocidad en el posicionamiento (Francés, 2005). Como consecuencia de esto, la toma de decisiones importantes y decisivas se hacen cada vez más frecuentes y exigentes, haciendo que la disponibilidad de información por adelantado represente la base fundamental para tomar la decisión correcta.

Normalmente, la toma de decisiones importantes desencadena un conjunto de actividades a realizar, provocando, de esta manera, el nacimiento de un proyecto. Con el nacimiento de un proyecto, viene implícito un nivel de incertidumbre, que según el área de conocimiento de aplicación, puede presentarse en menor o mayor grado. Dicha incertidumbre se presenta con mayor intensidad particularmente en los proyectos de tecnología, en los cuales aún no se tiene completa comprensión del problema a resolver, dificultando así el análisis de factibilidad (Palacios, 2005).

Durante los últimos 50 años, los sistemas de computadoras han introducido un nuevo orden o manera de realizar las cosas. Y sin importar lo antiguo que puedan ser las palabras de Maquiavelo hace 500 años: “No hay nada más difícil de llevar a cabo, más peligroso de realizar o de éxito más incierto que encabezar la introducción de un nuevo orden de cosas”

(c.p. Pressman, 2006), dichas palabras aún tienen vigencia y es perfectamente aplicable a la tecnología de la información.

En el año 2003, Piorun, en su artículo intitulado “¿Por qué fracasan los proyectos?”, expresa la debilidad que existe en la estimación de esfuerzo y tiempo en la actualidad y la importancia en profundizar el trabajo de métricas. El autor añade además, que esta temática es muy clara en algunas especialidades como la construcción, mientras que con los proyectos de tecnología de información sucede todo lo contrario. Según Pressman (1998), esta problemática representa uno de los factores que contribuye a lo que se llamó en alguna oportunidad “la Crisis del Software”; sin embargo, este autor aclara que la situación se ha extendido por tanto tiempo que sería más adecuado llamarlo “Aflicción Crónica”.

Según Reel (1999) (c.p. Pressman, 2001, p. 39) “en 1998, los datos de la industria del software indicaron que el 26 por ciento de proyectos de software fallaron completamente y que el 46 por ciento experimentaron un desbordamiento en la planificación y en el coste”. Siendo 46% un porcentaje extremadamente alto y alarmante, queda completamente identificada la necesidad de métodos de estimación de proyectos de software más adecuados a la realidad.

En el estudio “Soft Cost Estimation” (Leung & Fan, 2006), los autores enumeran varios métodos de estimación, clasificados principalmente por métodos algorítmicos y no algorítmicos. Los métodos algorítmicos están basados en modelos matemáticos que producen una estimación de costo o esfuerzo como una función de un número de variables (Esfuerzo =  $f(x_1, x_2, \dots, x_n)$ ); entre ellos, se puede nombrar el Punto de Función (PF) y el Modelo Constructivo de Costes (COCOMO por su acrónimo del inglés Constructive Cost Model).

Según Pressman (2006), PF parte de una matriz donde se debe identificar cuantitativamente y cualitativamente (Simple, Promedio, Complejo) la cantidad de funcionalidades que tendrá el sistema, las cuales se clasifican como: entradas externas, salidas externas, consultas externas, archivos de lógica interna y archivos de interfaz externa. Una vez identificadas dichas funcionalidades y mediante un conjunto de ecuaciones, se determina cuántos PF tendrá el

software que se desea desarrollar. Posteriormente, mediante datos históricos de la compañía, se podrá determinar a cuánto tiempo equivale cada PF, logrando así una estimación de horas/hombre necesarias para el desarrollo del proyecto de software.

Por otro lado, el modelo COCOMO, según Pressman (2006), ha evolucionado en los últimos años llegando a lo que se ha denominado COCOMO II. Dicha técnica tiene mucha similitud con el modelo PF en cuanto se debe identificar el número de pantallas, reportes y componentes que probablemente se requiera para construir la aplicación, clasificándolos en 3 niveles de complejidad: simple, medio o difícil. Finalmente, mediante ecuaciones matemáticas se obtiene el número de horas/hombre estimadas que se deberá emplear para el desarrollo del proyecto.

En un estudio de Estimación de Esfuerzo en Proyectos de Software utilizando Analogías (Shepperd & Schofield, 1997), los autores afirman que en esas dos últimas décadas se desarrollaron actividades en el área de predicción del esfuerzo con soluciones tipificadas por ser algorítmicas; sin embargo, esfuerzos por descifrar técnicas no algorítmicas se han venido desarrollando últimamente.

Las técnicas no algorítmicas, como su nombre lo dice, se caracterizan por carecer de modelos matemáticos encargados de generar la estimación. Leung y Fan (2006) enumeran las técnicas no algorítmicas como Parkinson, Bottom-Up, Top-Down, Analogía y Juicio del Experto.

En el método Parkinson, el costo es determinado por los recursos disponibles y no por la evaluación de los objetivos; es decir, si el software debe ser entregado en 12 meses y solo existen 5 personas disponibles para el trabajo, el esfuerzo necesario es de 60 personas/mes. Obviamente esto no promueve una buena práctica de ingeniería de software.

Por otro lado, las técnicas Bottom-Up (Abajo-Arriba) y Top-Down (Arriba-Abajo) consisten en la desagregación del trabajo, similar al WBS (de sus siglas en inglés Work

Breakdown Structure) donde la estimación del costo se realizará de abajo hacia arriba o de arriba hacia abajo según sea el caso.

Continuando con las técnicas no algorítmicas, el Juicio del Experto consiste en consultar uno o más expertos, quienes utilizarán sus propios métodos y experiencias para hacer la estimación. Y por último, la técnica por Analogía consiste en comparar el proyecto actual con uno similar y mediante el razonamiento por analogía lograr una estimación. Según Leung y Fan (2006), en ambos casos se requiere información previa sobre proyectos pasados, concepto en que se sustentará el presente estudio; sin embargo, se aclara que es posible conseguir autores como Crespo (2003), que difiere en que la técnica por analogía usa la experiencia de los expertos como fuente esencial de conocimiento.

Particularmente, en la técnica por analogía y resaltando la necesidad de registros de proyectos pasados, Pressman (1998) expresa que: “Las métricas recopiladas de proyectos anteriores se utilizan como una base desde la que se realizan las estimaciones del esfuerzo y del tiempo para el actual trabajo de software” (p. 55). Piorun (2003) confirma lo anteriormente dicho, cuando expresa la necesidad de “disponer, o comenzar a construir, una base de conocimiento de la organización, que se alimentará con las experiencias de proyectos que se vayan desarrollando” (p. 1).

En este mismo orden de ideas, el grupo de estándares internacionales de software ISBSG (de sus siglas en inglés International Software Benchmarking Standards Group) empezó a estandarizar la recolección de datos históricos mediante formatos de planillas disponibles en su página Web. Así mismo, este grupo ha dado el paso adicional de recopilar dicha información desde el año 1997 y crear una Base de Datos Internacional, que cuenta en la actualidad con información de más de 4.100 proyectos de software.

Retomando la técnica de analogía y, quedando clara la disposición de una gran magnitud de “datos históricos”, se hace evidente que la capacidad humana de búsqueda de proyectos similares y su posterior estimación por analogía, se ve limitada ante el volumen de información que se debe consultar y la tendencia a la subjetividad que tiene el hombre por

naturaleza. Es así como toma importancia el desarrollo de técnicas automatizadas que puedan procesar el conocimiento de proyectos pasados y compararlos con nuevos proyectos mediante un razonamiento lógico, para lograr una estimación conveniente. Esto constituye en la actualidad un nuevo campo de estudio por los retos que implica.

En este sentido, lograr un razonamiento que busque similitudes entre dos objetos de estudio para predecir un comportamiento, es una capacidad humana que la Inteligencia Artificial (IA) ha venido estudiando en los últimos años. “El término "inteligencia artificial" se acuñó en Dartmouth, en 1956, durante una conferencia convocada por McCarthy” (Bargiela, 2007, p. 7). Barr y Feigenbaum (1981) (c.p Bargiela, 2007, p.8) definen la IA, como “la parte de la ciencia que se ocupa del diseño de sistemas de computación inteligentes; es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la comprensión del lenguaje, el aprendizaje, el razonamiento, la resolución de problemas, etc.”

Partiendo de la IA como herramienta para resolver la estimación de proyectos por analogía, han surgido diferentes ramas de investigación. En este sentido, Crespo (2003) en su trabajo monográfico, enumera las técnicas utilizadas de la siguiente manera: redes neuronales, técnicas orientadas al aprendizaje, razonamiento basado en casos, árboles de regresión, lógica difusa, agentes dinámicos y programación genética. Cada una de las técnicas requiere una dedicación especial para su entendimiento; sin embargo, para efectos del presente estudio, se le prestará especial atención a la técnica de Redes Neuronales Artificiales (RNA).

Aggarwal, Singh, Chandra & Puri (2005) en su trabajo “Bayesian regularization in a neural network model to estimate lines of code using function points” justifican la utilización de RNAs enumerando las capacidades que éstas poseen:

- 1) Pueden modelar complejas relaciones no lineales y aproximar cualquier función de medida.
- 2) Pueden ser usadas como una herramienta efectiva para la clasificación y ordenamiento de patrones.

- 3) Son particularmente útiles cuando hay una compleja relación entre las entradas y salidas.

Las RNAs son una abstracción simplificada de lo que representa una red neuronal natural, en donde un grupo de neuronas recibe un estímulo (entrada) y mediante su red de interconexión van transmitiendo el estímulo hasta generar una reacción (salida). En la aplicación de las RNAs para el proceso de estimación, las entradas (estímulos o parámetros) de la red lo constituyen un conjunto de características del proyecto; como salida se busca obtener la estimación (de costo o de esfuerzo) que se desea obtener.

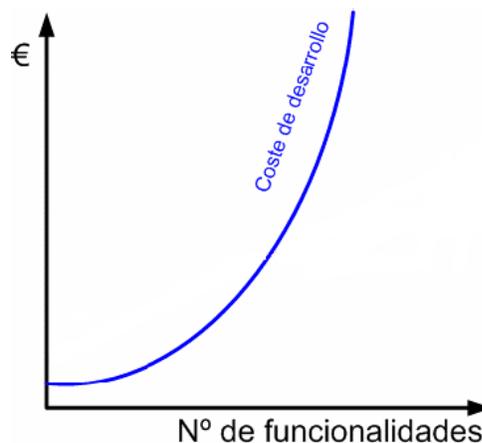
En este mismo orden de ideas, diferentes investigaciones se han desarrollado en este campo. En 1994, Wittig y Finnic en su trabajo “Using artificial neural networks and function points to estimate 4gl software development effort” aplicaron RNAs para estimar el esfuerzo en horas/hombre partiendo de una etapa intermedia del proceso del cálculo de PF (PF sin ajuste). Este estudio se basó en una base de datos de 15 proyectos, concluyendo que el modelo es capaz de estimar el esfuerzo en horas/hombre (número de horas necesarias para construir un software) con un promedio de error del 10%.

En el estudio realizado por Aggarwal et al. (2005), los autores utilizan las RNAs para la estimación del número de líneas de código partiendo de PF utilizando diferentes métodos de aprendizaje. En este mismo estudio se toma información histórica de la base de datos ISBSG versión 9 y sus autores concluyen que las RNAs pueden ser usadas para obtener resultados muy aproximados a la realidad de los hechos, una vez que los PF son conocidos.

Si bien es cierto que Aggarwal et al. (2005) llegaron a un resultado satisfactorio, es conveniente mencionar que en la etapa inicial del proyecto de software, la comprensión de los requerimientos no es algo común, complicando el conocimiento de los PF. Como soporte a esta información, “Christel y Kang (1992) identifican una serie de problemas que ayudan a entender por qué es difícil la obtención de requisitos:

- ◆ Problemas de ámbito: El límite del sistema está mal definido o los clientes/usuarios especifican detalles técnicos innecesarios que pueden confundir, en lugar de clarificar los objetivos generales del sistema.
- ◆ Problemas de comprensión: Los clientes/usuarios no están seguros por completo de qué es lo que necesitan, comprenden poco acerca de las capacidades y limitaciones de su ambiente de cómputo, no comprenden del todo el dominio del problema, tienen dificultades al comunicar necesidades al ingeniero de sistemas, omiten información que consideran “obvia”, especifican requisitos que chocan con la necesidad de otros clientes/usuarios, o especifican requisitos ambiguos o inestables.
- ◆ Problemas de volatilidad: Los problemas cambian conforme transcurre el tiempo.” (Pressman, 2006, pp. 158-159).

Lo dicho anteriormente confirma el hecho de que el escenario inicial es susceptible a cambio, a medida que el proyecto se desarrolla. Estos cambios hacen variar el escenario aumentando o disminuyendo la complejidad estimada inicialmente. Según Ordieres, Torralba y Chiner (2004), a medida que el número de funcionalidades aumenta, el costo de desarrollo también aumenta, pero no de manera lineal, sino más bien de manera exponencial (ver Gráfica 1).

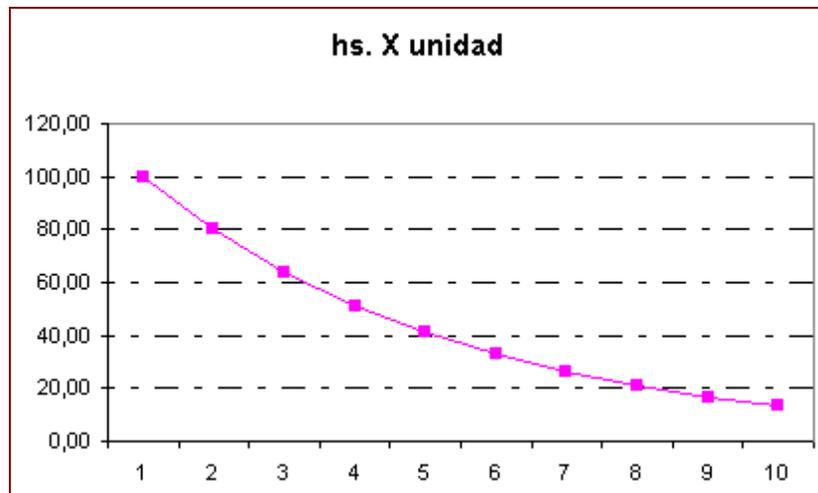


Gráfica 1. Efecto más realista de los costes de desarrollo y valor aportado por el producto del proyecto frente a las funcionalidades. (Ordieres et al., 2004).

En vista de que el costo de desarrollo está directamente relacionado con el número de horas/hombre necesarias para ello, es posible deducir que una gráfica que represente el número de funcionalidades y el número de horas/hombres necesarias para el desarrollo, siga el mismo patrón de comportamiento.

Por otro lado, otro factor de gran influencia en las estimaciones, es el grado de experiencia del equipo de trabajo y el gerente de proyecto. La experiencia individual de las personas que trabajan en proyectos en diferentes roles, afectan la productividad total del equipo, reflejándose positivamente o negativamente en variables de tiempo, costos y calidad.

Este último tema está relacionado con las curvas de aprendizaje (ver Gráfica 2), lo cual no es más que una línea que muestra la relación existente entre el tiempo (o costo) de producción por unidad y el número de unidades de producción consecutivas (Lefcovich, 2006).



Gráfica 2. Horas/Hombre x Unidad Producida (Lefcovich, 2006).

Con base en este conocimiento, se puede hacer la similitud con la experiencia ganada por los integrantes de equipos de proyectos para el desarrollo de los mismos. Esperando así que a medida que se vaya aumentando la experiencia, el tiempo de desarrollo del software sea menor.

Por lo antes expuesto, el uso de RNAs entrenadas con datos de proyectos reales para la estimación de tiempo y esfuerzo, teniendo como entrada los años de experiencia y los puntos

de función (conteo del número de entradas externas, salidas externas, consultas externas, archivos de lógica interna y archivos de interfaz externa), plantean la siguiente inquietud: ¿será la RNA capaz de detectar la relación existente en entradas como la experiencia y las entradas de PF, y el esfuerzo necesario para el desarrollo?.

Con base en el tema abierto antes expresado, se deriva la motivación del presente estudio, el cual consiste en un planteamiento para variar de manera controlada la experiencia y los puntos de función del software y, por medio de una RNA, fundada en información histórica de proyectos provenientes de una base de datos, analizar en qué proporción afecta el esfuerzo (en horas/hombres) necesario para desarrollar un proyecto de software.

El objetivo perseguido es que el resultado de este análisis contribuya al mejor entendimiento del comportamiento de los proyectos de software a medida que ciertas condiciones cambian basándose en datos reales. Así mismo, se podrá observar el grado de influencia que tienen estas variables de entrada, según la realidad de los proyectos, dando pie a futuras comparaciones con otros modelos de estimación.

## **2. OBJETIVOS DE LA INVESTIGACION**

### **2.1.Objetivo General**

- Realizar un análisis del efecto de las variables experiencia del equipo y los puntos de función, sobre el esfuerzo necesario para el desarrollo de proyectos de software mediante RNAs entrenadas con una base de datos de históricos de proyectos.

## **2.2.Objetivos Específicos**

- Estudiar la estimación de proyectos de software mediante técnicas automatizadas, en particular, el uso de RNAs.
  
- Determinar la arquitectura de RNA a utilizar.
  
- Entrenar la RNA utilizando una base de datos histórica de proyectos.
  
- Analizar el comportamiento del esfuerzo en relación con la experiencia del equipo de proyecto.
  
- Analizar el comportamiento del esfuerzo en relación con los puntos de función.

## **CAPÍTULO II.**

### **MARCO CONCEPTUAL**

El presente capítulo proporciona al lector los conocimientos básicos necesarios para la comprensión del contexto en que se maneja este trabajo de investigación. Inicia con conceptos generales como lo son proyectos de software y sus estimaciones, para luego entrar en las diferentes técnicas de estimación en donde se destacan las técnicas relacionadas con I.A., dando especial atención a las que utilizan RNAs.

#### **1. PROYECTOS DE SOFTWARE**

Según Palacios (2005), un proyecto se puede definir de la siguiente manera:

“Un proyecto es un trabajo que realiza la organización con el objetivo de dirigirse hacia una situación deseada. Se define como un conjunto de actividades orientadas a un fin común, que tiene un comienzo y una terminación. Las características fundamentales de un proyecto son la temporalidad del trabajo y el resultado final que es un producto o servicio único” (p. 17)

Dependiendo del producto que se requiere obtener con el proyecto, el mismo se desenvolverá entre las diferentes áreas del conocimiento humano (civiles, médicos, sociales, etc.). Ahora bien, en el caso particular de los proyectos de software, que es el área específica donde se centra este estudio, el producto final obtenido es un software. En este sentido, Pressman (2006) desglosa el producto de software en tres subproductos:

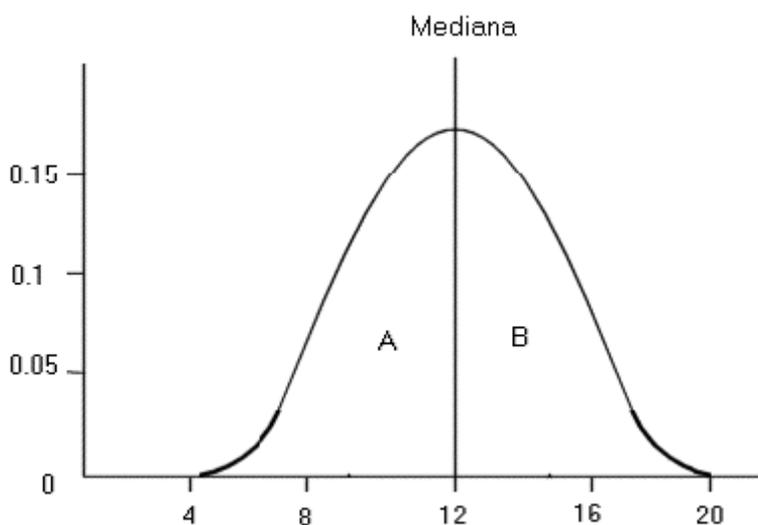
- 1) Las instrucciones (programas de computadora) que al ejecutarse proporcionan las características, funciones y el grado de desempeño deseado.
- 2) Las estructuras de datos que permiten que los programas manipulen información de manera adecuada.
- 3) Los documentos que describen la operación y el uso de los programas.

## 2. ESTIMACIÓN DE PROYECTOS DE SOFTWARE

Tal como lo expresa Palacios (2005), en la vida se presentan oportunidades que requieren de la toma de decisiones objetivas. Este proceso se presenta tanto en el ámbito personal como en el organizacional; sin embargo en este último, se hace más estricto el uso de metodologías de evaluación y decisión, fundamentadas en hechos, con datos cuantificados estadísticamente para limitar la subjetividad. Junto con la toma de una decisión, pueden generarse un conjunto de actividades dando paso al nacimiento de un proyecto.

Este estudio que se realiza previo al desarrollo del proyecto es denominado “Formulación”. En este proceso se encuentra el estudio de mercado, técnico y financiero, que cierra finalmente con la evaluación donde se decide si se llevará a cabo el proyecto.

En el proceso de formulación, también conocido como “factibilidad”, toma fundamental importancia la estimación. Cerrillo (1999) explica que la estimación debe comprenderse como un rango más que como un simple número, no es un objetivo sino una valoración probabilística donde la valoración obtenida representa el centro del rango probable de valores (ver Gráfica 3).



Gráfica 3. Número de meses para completar el proyecto (Cerrillo, 1999).

Una vez que el proyecto se ha terminado, se tiene la oportunidad de comparar los valores reales con los estimados. Si  $V_F$  es el valor estimado y  $V_A$  es el valor real, el error relativo viene representado por la expresión (1).

$$MRE = \frac{|V_A - V_F|}{V_A} \quad (1)$$

Con frecuencia se necesita obtener el error relativo de un conjunto de estimadores. Por ejemplo, usualmente se quiere saber si las predicciones de esfuerzo son precisas para un grupo de proyectos desarrollados. De esta manera surge el error relativo medio para  $n$  proyectos (ver expresión (2)).

$$\overline{RE} = \frac{1}{n} \sum_{i=1}^n RE_i \quad (2)$$

Los rangos probables de valores para la estimación dependen de la etapa en que se encuentra el proyecto. La Asociación por el Avance de la Estimación de Costo (AACE The Association for the Advancement of Cost Estimation), en Febrero del 2005, publicó la mejor práctica No. 18R-97 en donde se puede observar una matriz de clasificación de las estimaciones de costo según la etapa en que se encuentra el proyecto (ver Tabla 1).

Tabla 1. Matriz de Clasificación de Estimados de Costos para Procesos Industriales. (AACE International, 2005).

Estimado de Clase	Característica Primaria	Característica Secundaria		
		Uso Final: Propósito típico de la estimación	Metodología: Método típico de estimación	Rango esperado de Precisión: Típica Variación entre dos rangos
Clase 5	0% to 2%	Sondeo	Factor de Capacidad, Modelos Paramétricos, Juicio de Expertos, o Analogía	Baja: -20% a -50% Alta: 30% a 100%
Clase 4	1% a 15%	Estudio o Factibilidad	Factor de Equipos o Modelos Paramétricos	Baja: -15% a -30% Alta: 30% a 100%
Clase 3	10% a 40%	Presupuesto, Autorización o Control	Lista semidetallada de Ítems	Baja: -10% a -20% Alta: 10% a 30%
Clase 2	30% a 70%	Control u Ofertas	Lista de Costo Detallado con Recibos Detallados Forzados	Baja: -5% a -15% Alta: 5% a 20%
Clase 1	50% a 100%	Chequeo de la Estimación	Lista de Costo Detallado con Recibos Detallados	Baja: -3% a -10% Alta: 3% a 15%

Dado que la formulación es realizada bajo un nivel de definición de proyecto muy bajo, se le presta mayor interés a las estimaciones de Clase 5 y Clase 4.

### **Estimación Clase 5:**

Está preparado generalmente con base en una información muy limitada, y por consecuencia tiene un amplio margen de error. Algunas compañías y organizaciones han decidido no considerar esta clasificación como convencional debido al grado de error. Debido a los limitados requerimientos de uso final, las estimaciones clase 5 pueden ser preparadas en corto tiempo y con poco esfuerzo. El uso final es para la preparación de planes de negocios, evaluación o viabilidad inicial, evaluación de esquemas alternativos, sondeo de proyectos, estudio de localización de proyectos, evaluación de necesidades de recursos y presupuesto, etc.

#### **Estimación Clase 4:**

Al igual que las estimaciones de clase 5, las de clase 4 son normalmente basadas en información limitada y por consiguiente aún mantienen un amplio margen de error. Es utilizado para determinar factibilidad, evaluación de conceptos y aprobación de un presupuesto inicial. Normalmente aplicado cuando la ingeniería está completada en un porcentaje entre 1 y 15%, incluye la creación de algunos diagramas de procesos principales, capacidad de planta, ingeniería preliminar de procesos y lista de equipos necesarios. Es utilizado para el planeamiento estratégico detallado, desarrollo de negocios, sondeo de proyectos en estados más avanzados, análisis de esquemas alternativos, confirmación de factibilidad técnica y/o económica y aprobación del presupuesto inicial o aprobación para proceder al siguiente paso.

Enfocándose más en la estimación en términos de software, Pressman (2006) lo define como: “intento por determinar cuánto dinero, esfuerzo, recursos y tiempo tomará construir un sistema o producto específico basado en software” (p. 691).

El proceso de estimación es bastante delicado debido al alto nivel de incertidumbre y al riesgo implícito. Los hechos que dificultan el proceso se enumeran a continuación:

- 1) No existe un modelo de estimación universal o una fórmula que pueda ser usada para todas las organizaciones.
- 2) Hay muchas personas implicadas en los proyectos que necesitan de estimaciones.
- 3) Generalmente, la estimación se hace superficialmente, sin apreciar el esfuerzo requerido para hacer un trabajo.
- 4) Las estimaciones claras, completas y precisas son difíciles de formular, especialmente al inicio del proyecto.
- 5) La rapidez con la que cambia la tecnología de la información y las metodologías de desarrollo de software son un problema para la estabilización del proceso de estimación.

Todas estas dificultades no son razón para desistir en la búsqueda de estimaciones precisas en etapas tempranas de la formulación de proyectos de software. En este sentido, Leung y Fan (2006) enumeran su importancia de la siguiente manera:

- 1) Ayuda a clasificar y priorizar el desarrollo de los proyectos con respecto a la generalidad del plan de negocios.
- 2) Ayuda a determinar qué recursos serán usados para llevar a cabo el proyecto y su buen su aprovechamiento.
- 3) Ayuda a evaluar el impacto de cambios.
- 4) Los proyectos son más fáciles de manejar y controlar cuando los recursos están mejor ajustados a las necesidades reales.
- 5) Los clientes esperan costos de desarrollo reales para estar alineados con la estimación de costos.

### **3. MÉTODOS DE ESTIMACIÓN DE PROYECTOS DE SOFTWARE**

El conocimiento de las diferentes formas de clasificar los métodos de estimación de proyectos, proporciona una idea de las características diferenciadores que éstos poseen, permitiendo la mayor comprensión de un método en particular, a la hora de definirlo.

Son muchas las clasificaciones que se pueden encontrar; Crespo (2003) hace un recorrido entre éstas, donde resalta en primera instancia la existencia de métodos manuales y métodos automáticos.

- ◆ **Métodos Manuales:** Estos métodos no utilizan herramientas de cálculo automático y su fundamento conceptual son las “reglas de oro” (rule of thumb). Es un principio con amplia utilización y que no intenta ser estrictamente fiable y confiable para todas las situaciones. Es fácil de aprender y aplicar para cálculos aproximados de valores.

- ◆ **Métodos Automáticos:** Estos métodos han sido implementados en herramientas automáticas que les permiten proporcionar capacidades que no pueden dar los métodos manuales. Como base de sus cálculos suelen emplear bases de datos de proyectos históricos que le sirven para ajustar las ecuaciones de estimación. Esto les permite obtener una respuesta más afinada en la estimación de nuevos proyectos.

Crespo comenta además, tres niveles de aplicación para cada uno de los métodos anteriormente señalados:

- 1) **Nivel de proyecto:** Cuando la estimación es aplicada a la totalidad del software que se desea construir.
- 2) **Nivel de fase:** Cada fase recibe su propia estimación partiendo de porcentajes de la totalidad del proyecto.
- 3) **Nivel de actividad o tarea:** Las estimaciones se hacen para cada actividad o tarea del trayecto, las cuales han sido determinadas al principio del mismo, utilizando técnicas como “Work Breakdown Structure” (WBS). Este es el nivel de estimación de mayor precisión.

Adicionalmente, este autor promueve una nueva clasificación que se resume de la siguiente manera:

- ◆ **Modelos matemáticos paramétricos:** Son aquellos modelos que utilizan ecuaciones matemáticas para realizar las estimaciones. No se introduce la diferencia de si han sido generadas por expertos porque se supone que la intervención de expertos es imprescindible en el desarrollo de la ecuación. Tampoco se diferencia entre modelos tabulares, multiplicativos, lineales y no lineales. Los motivos son que, por una parte, las ecuaciones multiplicativas y lineales se consideran un caso particular de las no lineales y, por otra, los modelos tabulares o los que utilizaban ecuaciones lineales están ya en desuso y todos los nuevos modelos publicados utilizan ecuaciones no lineales.

- ◆ **Estimación basada en analogías:** Esta técnica se basa en la comparación de los resultados reales de diferentes proyectos entregados a cliente. Este método será tocado con mayor profundidad más adelante.
- ◆ **Modelos dinámicos:** Basados en la variabilidad de los factores de coste y duración de un proyecto de software durante su desarrollo. Estos modelos consideran a priori que la variabilidad que se produce es suficientemente grande como para afectar de forma considerable a los resultados estimados del proceso.
- ◆ **Modelos basados en técnicas de I.A.:** Estos métodos se basan en el uso de datos recogidos en proyectos anteriores en los que se realizaron estimaciones y usando diferentes técnicas de extracción de conocimiento de inteligencia artificial.

Esta última clasificación proporcionada por Crespo (2003) tiene mucha similitud con la utilizada por Leung y Fan (2006), donde expresa básicamente dos vertientes principales: Métodos Algorítmicos y Métodos no Algorítmicos (conformados por el método de analogía, modelos dinámicos y los modelos de inteligencia artificial).

### 3.1.Métodos Algorítmicos

Esta clasificación, propuesta por Leung y Fan (2006), coincide con los modelos matemáticos paramétricos de Crespo (2003). En ella, se logra proporcionar una estimación como una función de un número de variables. Todo modelo algorítmico tiene la expresión (3).

$$\text{Esfuerzo} = f(x_1, x_2, \dots, x_n) \quad (3)$$

Donde  $\{x_1, x_2, \dots, x_n\}$  denotan factores de costo. Los métodos algorítmicos existentes difieren en dos aspectos: factores de costo y la forma de la función  $f$ .

1) **Factores de costo:** Además del tamaño del software, hay otros factores que intervienen en la estimación:

- ◆ **Factores del productos:** Seguridad requerida, complejidad del producto, tamaño de la base de datos, reusabilidad requerida, documentación acorde a las necesidades del ciclo de vida.
- ◆ **Factores de computación:** Restricciones de tiempo de ejecución, restricciones de almacenamiento, volatibilidad de la plataforma.
- ◆ **Factores del personal:** Capacidad de análisis y programación, experiencia, conocimiento de la plataforma, del lenguaje y las herramientas, continuidad del personal.
- ◆ **Factores del proyecto:** Sitios múltiples de desarrollos, uso de herramientas de software, horario de desarrollo requerido.

Los factores antes nombrados no son necesariamente independientes y muchos de ellos son difíciles de cuantificar. En muchos modelos, algunos factores aparecen combinados o son simplemente ignorados.

2) **Forma de la función F:**

- ◆ **Modelo lineal:** En la expresión (4), el coeficiente  $a_1, \dots, a_n$  son seleccionados para ajustar la información del proyecto.

$$\text{Effort} = a_0 + \sum_{i=1}^n a_i x_i \quad (4)$$

- ◆ **Modelo multiplicativo:** En la expresión (5), cada  $x_i$  toma 3 posibles valores  $-1, 0, +1$ .

$$\text{Effort} = a_0 \prod_{i=1}^n a_i^{x_i} \quad (5)$$

- ◆ **Modelo exponencial:** En la siguiente ecuación (expresión (6)), S es el tamaño del código, a y b son usualmente funciones de otros factores de costo. En este modelo se encuentra el COCOMO.

$$\text{Effort} = a \times S^b \quad (6)$$

- ◆ **Modelo de calibración usando regresión lineal:** Welti (2002) explica este método de la siguiente manera:

“Considérese la relación dada en (7) para explicar el comportamiento de una variable dependiente (Y) en función de  $n$  variables dependientes, donde  $f()$  es una forma funcional implícita.

$$Y = f(X_1, X_2, \dots, X_n) \quad (7)$$

En el caso en el cual esta forma funcional no pueda estimarse,  $f()$  puede aproximarse según la expresión dada en (8).

$$Y = \sum_{i=1}^n \beta_{i+1} X_i + \Psi \quad (8)$$

Para  $i = 1, 2, \dots, n$  donde las  $\beta$  son parámetros de la función y  $\Psi$  es el error debido a la aproximación lineal de (7)” (p. 3)

- ◆ **Modelo Discreto:** Tiene una forma tabular en el cual, usualmente relaciona el esfuerzo, la duración, la dificultad y otros factores. Tanto Leung y Fan (2006) como Crespo (2003), coinciden que este modelo tuvo cierta popularidad a los inicios de las estimaciones de costo, cuando eran fáciles de usar; sin embargo, actualmente se encuentra en desuso.

Dentro de los métodos algorítmicos se pueden mencionar el Punto de Función y COCOMO. Es importante resaltar que, para los efectos de la presente investigación, se presta fundamental atención a la manera de obtener los datos para la posterior medición del tamaño del software.

### 3.1.1. Punto de Función

Pressman (2006) se basa en el método propuesto inicialmente por Albretch (1979), el cual sirve para medir de manera efectiva la funcionalidad que entrega un sistema. Empleando datos históricos, el PF se usa para:

- 1) Estimar el costo o el esfuerzo requerido para diseñar, codificar y probar el software.
- 2) Predecir el número de errores encontrados durante la prueba.
- 3) Pronosticar el número de componentes y/o líneas de códigos.

El proceso se inicia con la medición del dominio de información, el cual se define de la siguiente manera:

- ◆ **Número de entradas externas:** Se origina en un usuario o es transmitida desde otra aplicación y proporciona distintos datos orientados a la aplicación o información de control.
- ◆ **Número de salidas externas:** Se deriva en el interior de la aplicación y proporciona información al usuario (informes, pantallas, mensajes, sonidos).
- ◆ **Número de consultas externas:** Entrada en línea que lleva a la generación de alguna respuesta inmediata por parte del software.
- ◆ **Número de archivos lógicos internos:** Agrupamiento lógico de datos que residen en los límites de las aplicaciones.
- ◆ **Número de archivos de interfaz externos:** Agrupamiento lógico de datos externos a la aplicación pero que proporcionan datos que podrían ser útil.

Tabla 2. Cálculo de Puntos de Función. (Pressman, 2006).

Valor de dominio de información	Conteo		Factor de ponderación			Total
			Simple	Promedio	Complejo	
Entradas externas		X	3	4	6	=
Salidas externas		X	4	5	7	=
Consultas externas		X	3	4	6	=
Archivos de lógica interna		X	7	10	15	=
Archivos de interfaz externa		X	5	7	10	=

Una vez que se han recolectado los datos, se completa el formulario que se muestra en la Tabla 2 y se asocia un valor de complejidad a cada conteo.

Para calcular los PF se usa la relación dada en (9).

$$PF = \text{conteo total} \times [0.65 + 0.01 \times \sum(F_i)] \quad (9)$$

Donde conteo total es la suma de todas las entradas de PF obtenidas en la Tabla 2, también llamado **puntos de función sin ajustar**.

$F_i$  (i=1 a 14) son factores de ajuste de valor, con base en las respuestas a las siguientes preguntas (Pressman, 2006, p. 475):

- 1) ¿El sistema requiere respaldo y recuperación confiables?
- 2) ¿Se requieren comunicaciones de datos especializados para transferir información a la aplicación, u obtenerla de ella?
- 3) ¿Hay funciones distribuidas de procesamiento?
- 4) ¿El desempeño es crítico?
- 5) ¿El sistema se ejecutará en un entorno existente que tiene un uso pesado de operaciones?
- 6) ¿El sistema requiere entrada de datos en línea?
- 7) ¿La entrada de datos en línea requiere que la transacción de entrada se construya en varias pantallas u operaciones?
- 8) ¿Los archivos de lógica interna se actualizan en línea?
- 9) ¿Las entradas, las salidas, los archivos o las consultas son complejos?
- 10) ¿Es complejo el procesamiento interno?
- 11) ¿El código diseñado será reutilizable?
- 12) ¿Se incluyen la conversión e instalación en el diseño?
- 13) ¿Está diseñado el sistema para instalaciones múltiples o diferentes organizaciones?
- 14) ¿Aplicación diseñada para cambio?

A cada pregunta se le asigna un valor de 0 a 5, donde 0 significa que no aplica y 5 significa que es fundamental. Posteriormente, los PF son comparados con los proyectos pasados donde se establece una correspondencia entre PF y tiempo de desarrollo, tiempo de prueba, costo, etc.

### 3.1.2. *Modelo COCOMO*

El modelo aborda las siguientes áreas:

- **Modelo de composición de la aplicación:** Utilizado en primeras etapas donde son prioridad la elaboración de prototipos y consideraciones de interacción, desempeño y madurez de la tecnología.
- **Modelo de etapa de diseño temprano:** Utilizado cuando se tienen los requisitos y la arquitectura básica de software.
- **Modelo de etapa posterior a la arquitectura:** Usado durante la construcción del software.

Como punto de partida y, el de mayor interés para el presente estudio, se debe estimar el tamaño del software a realizar, los cuales parten de tres opciones: PF (nombrado previamente), puntos de objeto y líneas de código fuente.

Los puntos de objetos se calculan mediante conteos de pantallas, reportes y componentes que se requieran construir. Una vez determinados estos elementos, se multiplican por los factores de ponderación y se totaliza (ver Tabla 3).

Tabla 3. Ponderación de complejidad para tipos de objeto. (Boehm, 1996, c.p. Pressman, 2006)

Tipo de objeto	Peso de complejidad		
	Simple	Medio	Difícil
Pantalla	1	2	3
Reporte	2	5	8
Componente 3GL			10

Posteriormente se utiliza la ecuación (10) para obtener el número de puntos de objeto (NPO), donde “puntos objeto” representa el resultado de la tabla y “%reut” representa el porcentaje de reutilización de los componentes dentro del software.

$$\text{NPO} = (\text{puntos objeto}) \times [(100 - \%reut)/100] \quad (10)$$

Luego se calcula la tasa de productividad (PROD) basándose en la Tabla 4 y según la ecuación dada en (11).

$$\text{PROD} = \text{NPO} / \text{persona-mes} \quad (11)$$

Tabla 4. Tasa de productividad por puntos objeto. (Boehm, 1996, c.p. Pressman, 2006)

<b>Experiencia/capacidad del desarrollador</b>	<b>Muy Baja</b>	<b>Baja</b>	<b>Nominal</b>	<b>Alta</b>	<b>Muy Alta</b>
<b>Madurez/capacidad del entorno</b>	Muy Baja	Baja	Nominal	Alta	Muy Alta
<b>TASA PRODUCTIVIDAD (PROD)</b>	4	7	13	25	50

Por último, el esfuerzo estimado será lo expresado en (12).

$$\text{Esfuerzo Estimado} = \text{NPO} / \text{PROD} \quad (12)$$

Por otro lado, las líneas de código dependerán del lenguaje de programación. Se han utilizado tablas de conversión de PF o NPO a líneas de código (LDC) basándose en proyectos pasados, sin embargo, en la actualidad, con el surgimiento de aplicaciones de desarrollo rápido; esta medida no representa con claridad el tamaño del software, en cuanto gran parte del código producido es generado automáticamente a partir de interfaces de solicitudes de especificaciones.

### **3.2.Métodos no Algorítmicos**

Continuando con la clasificación de Leung y Fan (2006), se encuentran los métodos no algorítmicos, los cuales se clasifican en:

#### ***3.2.1. Juicio del Experto***

Este método involucra la consulta de uno o más expertos. El experto provee un estimado usando sus propios métodos y experiencias. Mecanismos de consenso de expertos tal como la técnica Delphi o PERT son utilizadas para resolver las inconsistencias en la estimación. La técnica Delphi funciona de la siguiente manera:

- 1) El coordinador presenta cada experto con una especificación y una manera de estimar.
- 2) Cada experto llena la información en forma individual permitiendo preguntas al coordinador.
- 3) El coordinador prepara un resumen de todas las estimaciones de los expertos y solicita otra ronda en caso de que los expertos lo consideren conveniente.
- 4) Se repite el paso 2 y 3 si así lo amerita.

Una variación de esta técnica puede ser que el coordinador llama a una reunión para discutir los puntos específicos donde la estimación se diferencia ampliamente.

#### ***3.2.2. Parkinson***

Usa el principio de Parkinson: “el trabajo se expande hasta llenar el volumen disponible” (Leung y Fan, 2006, p. 2); el costo es determinado por los recursos disponibles y no está basado en una evaluación objetiva.

### ***3.2.3. Precio para Ganar***

Las estimaciones son realizadas para establecer el mejor precio para ganar el proyecto. La estimación está establecida por el presupuesto del cliente en vez de las funcionalidades del software.

### ***3.2.4. Bottom-Up***

En este caso, cada componente del sistema es estimado separadamente y el resultado es sumado para obtener el estimado total. Este método requiere que el diseño inicial indique la descomposición del sistema en diferentes componentes.

### ***3.2.5. Top-Down***

Este método es lo opuesto al Bottom-Up. La estimación total del sistema es derivada de propiedades globales usando métodos algorítmicos o no algorítmicos, posteriormente dichas estimaciones son repartidas en cada uno de los componentes. Esta estimación es apropiada para escenarios iniciales de estimación.

### ***3.2.6. Estimación por Analogía***

Según Leung y Fan (2006), este método requiere uno o más proyectos completos que sean similares al nuevo proyecto, de manera de obtener la estimación mediante un razonamiento por analogía usando la estimación real del proyecto previo. La estimación por analogía puede ser hecha ya sea de un proyecto total como de un nivel del mismo.

La estimación por Analogía ofrece algunas ventajas comentadas por Shepperd y Schofield (1997):

- ◆ Evita los problemas asociados con la extracción y codificación del conocimiento.
- ◆ Solo necesitan lidiar con aquellos problemas que realmente ocurren en la práctica, mientras que los sistemas algorítmicos deben manejar todos los posibles problemas.

- ◆ Puede manejar casos en que la estimación inicial no fue correcta. Esto es útil pues habilita al usuario de identificar situaciones potenciales de riesgo.
- ◆ Es capaz de lidiar con un escaso dominio del problema, tal como ocurre en proyectos de software en las etapas iniciales.
- ◆ Los usuarios pudieran estar más deseosos de aceptar soluciones de sistemas basados en analogía, pues estos son derivados de una forma de razonamiento más asociado a la manera en que el humano soluciona los problemas.

#### **4. BASE DE DATOS HISTÓRICOS DE PROYECTOS**

Como es evidente, la disponibilidad de información histórica tiene una fuerte influencia en la estimación, por lo que garantizar su existencia se hace indispensable (Pressman, 2006).

Las empresas de desarrollo de software han hecho esfuerzos independientes por registrar información histórica de sus proyectos. Sin embargo, a la hora de comparar proyectos entre compañías, la diferencia en la manera de registrar la información histórica representa un inconveniente. El grupo de estándares internacionales de software ISBSG (de sus siglas en inglés International Software Benchmarking Standards Group), desde el año 1997 se ha encargado de proporcionar estándares para el registro de la información histórica de proyectos en las compañías de desarrollo de software. Adicionalmente, empresas han facilitado información ya estandarizada de sus proyectos a ISBSG logrando conformar en la actualidad una base de datos con más de 4.100 proyectos.

El ISBSG pone disponible en línea, una herramienta de estimación de software basado en una técnica de I.A. la cual se describe en la Sección 2.5. Y así como ésta, muchas otras herramientas de I.A. están siendo estudiadas con el fin de procesar los datos provenientes de este repositorio.

Es importante resaltar que, en el proceso de estandarización utilizado por el ISBSG, los puntos de función y puntos de objetos han funcionado como herramientas útiles para el

levantamiento de información, ofreciendo tablas que permiten determinar las dimensiones del software que se desea desarrollar.

Dentro de las estadísticas que proporciona el ISBSG, se tiene que las proporciones de los 5 elementos de medición de los puntos de función se han mantenido relativamente estables en los últimos 12 años, en base a los siguientes valores:

Entradas externas = 29%

Salidas externas = 24%

Consultas externas = 15%

Interfaces de usuarios = 8%

Archivos Lógicos = 24%

Con base en estos valores, sugieren que mediante el conocimiento de uno de estos parámetros, es posible la estimación del resto y así completar el total de puntos de función de una aplicación.

La adquisición de la base de datos del ISBSG se ve limitada por un proceso burocrático, costoso y largo en términos de tiempo, por lo que se puede utilizar una base de datos alternativa disponible en Internet de nombre Desharnais. Dicha base de datos, proveniente inicialmente de la tesis de maestría de J. M. Desharnais 1989 en la Universidad de Montreal, fue publicada por “PROMISE Software Engineering” (Sayyad & Menzies, 2005). Este grupo posee un repositorio de datos y herramientas publicado, a fin de que investigadores puedan construir modelos de software predictivos que sean repetibles, verificables o refutables.

Esta base de datos posee 81 registros de proyectos de los cuales 4 (identificados con el número 38, 44, 66, 75) están incompletos, por lo que normalmente se utilizan 77 proyectos de esta base de datos.

Los atributos de la base de datos son:

- Número del Proyecto: Número consecutivo que identifica al proyecto como único.
- Experiencia del Equipo: Cantidad de años promedio del equipo de proyecto.
- Experiencia del Gerente: Cantidad de años de experiencia del gerente de proyectos.
- Año del Proyecto: Año en que el proyecto fue finalizado.
- Duración: Cantidad de meses que llevó su desarrollo.
- Esfuerzo: Cantidad de horas/hombres consumidas para su desarrollo.
- Transacciones: Conteo de los puntos de función no ajustados relacionados con las entradas, salidas y consultas.
- Entidades: Conteo de los puntos de función no ajustados relacionados con los archivos internos y externos.
- Puntos sin Ajustar: Representa la suma de las Transacciones y las Entidades.
- Ajuste: Factor de ajuste utilizado en el cálculo de los puntos de función (ver Puntos de Función).
- Puntos de Función: Cálculo final de puntos de función. Viene dado por la ecuación:  $PF = \text{Puntos Sin Ajustar} * (0.65 + 0.01 * \text{Ajuste})$
- Lenguaje: Categoría de lenguajes de programación por su nivel de complejidad en la programación. Mientras mayor sea el número, más simple la programación.

## **5. INTELIGENCIA ARTIFICIAL (I.A.)**

Bargiela (2007) presenta varios conceptos de Inteligencia Artificial que se citan a continuación:

“La Inteligencia Artificial es la parte de la ciencia que se ocupa del diseño de sistemas de computación inteligentes; es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la comprensión del lenguaje, el aprendizaje, el razonamiento, la resolución de problemas, etc.

Métodos y algoritmos que permitan comportarse a las computadoras de modo inteligente.

Estudio de las facultades mentales a través del uso de modelos computacionales.

Desarrollo de autómatas (máquinas construidas por el hombre) para simular el proceso deductivo o inductivo humano de adquisición y aplicación de conocimiento” (p. 5).

Básicamente la I.A. se implementa, en su gran mayoría, mediante modelos matemáticos de algún tipo, es decir, se trata de representar o simplificar un problema mediante una abstracción que se realiza para comprender o manejar algo complejo, obteniendo algo más simple y a la vez representable en un algoritmo computacional.

## **5.1. Aplicaciones**

Existen diferentes aspectos de la vida humana y áreas del continente científico en que es posible aplicar algunas o todas las técnicas de la I.A. para la resolución de problemas.

Según Bargiela (2007), “algunas de las aplicaciones de la I.A. son:

- 1) Percepción y reconocimiento de imágenes (formas y colores).
- 2) Comprensión, generación y traducción del lenguaje natural.
- 3) Control de robots.
- 4) Juegos: ajedrez, damas, del tipo estratégicos, simulación, etc.
- 5) Matemáticas: cálculo simbólico, demostración de teoremas.
- 6) Computación: verificación de programas, aprendizaje automático.
- 7) Ingeniería: diseño, detección de fallos, planificación de manufacturación.
- 8) Análisis científico.
- 9) Diagnóstico médico y técnico.
- 10) Análisis financiero, entre otros.” (p. 10)

## **5.2. Técnicas de I.A. aplicadas en la estimación del esfuerzo en el desarrollo de software**

Según Crespo (2003), la I.A. aplicada a la estimación del esfuerzo en el desarrollo de software representa un método que no pertenece a modelos matemáticos paramétricos, ni a estimación por analogía. Sin embargo, parece contradictorio que, aun y cuando la I.A. se base en modelos matemáticos paramétricos, éste no sea ubicado dentro de los “métodos algorítmicos”. Así mismo, aun y cuando este método de estimación busca el razonamiento para comprender el comportamiento de los proyectos pasados y generar una estimación buscando la similitud con proyectos nuevos, tampoco sea clasificado dentro de la “estimación por analogía”.

Lo cierto es que la I.A. aplicada al campo de la estimación, busca simular el razonamiento humano en la resolución de este tipo de problemas y donde se combina diferentes técnicas que hacen que su clasificación posiblemente no sea evidente.

Crespo (2003) clasifica los métodos de I.A. de la siguiente manera:

- 1) Técnicas Orientadas al Aprendizaje
- 2) Lógica difusa
- 3) Agentes dinámicos
- 4) Programación genética
- 5) Redes Neuronales

A continuación se describen cada uno de los métodos.

### ***5.2.1. Algoritmos Evolutivos o Algoritmos Genéticos (AG)***

Bargiela (2007) explica los algoritmos evolutivos de la siguiente manera:

“Los AG se han desarrollado con la intención de optimizar poblaciones o conjuntos de soluciones posibles de un determinado problema mediante la utilización de la genética biológica.” (p. 12)

Tal como la selección natural de Darwin permitió que las especies evolucionaran, este método busca que cada solución vaya evolucionando mediante alteraciones al azar (mutaciones), cruces o combinaciones con otras soluciones con lo cual se obtienen nuevas soluciones que pueden o no ser mejores que las anteriores. Este paso se repite muchas veces, y en cada generación, la mejora con respecto a la anterior puede ser insignificante, pero al final del proceso, los individuos (soluciones) de la población obtenida representan las mejores soluciones para el problema.

### ***5.2.2. Sistemas Expertos***

Son sistemas compuestos por dos partes (Bargiela, 2007):

- Motor de Inferencias: procedimiento lógico que sigue un experto en determinada área del conocimiento.
- Base de Conocimientos: conocimiento del experto, pudiendo ser uno sólo o un grupo de expertos dentro de la misma área, inclusive pueden ser libros especializados en el tema de interés.

Este tipo de sistemas se desarrolla con los objetivos de imitar el conocimiento del experto, y así contar con su asistencia en cualquier lugar en poco tiempo con respecto al tiempo que tarda en formarse un profesional humano incluyendo el tiempo de experiencia.

Existen diferentes tipos de sistemas expertos clasificados según el modelo que utilizan, siendo algunos de éstos los siguientes:

- 1) Basados en Reglas.
- 2) Basados en Probabilidad.
- 3) Basados en Redes Neuronales Artificiales.
- 4) Basados en Lógica Difusa.

### 5.2.3. *Agentes Inteligentes*

Bargiela (2007) define agentes inteligentes de la siguiente manera:

“Todo aquello que puede considerarse que percibe su ambiente mediante sensores y que responde o actúa en el ambiente por medio de efectores” (p. 12).

Martin Fischles y Oscar Firschein en 1987 describieron los atributos del agente inteligente de la siguiente manera:

- 1) Tener actitudes mentales tales como creencias e intenciones.
- 2) Tener la capacidad de obtener conocimiento, es decir, aprender.
- 3) Poder resolver problemas, incluso particionando problemas complejos en otros más simples.
- 4) Entender: Poseer la capacidad de crearle sentido, si es posible, a ideas ambiguas o contradictorias.
- 5) Planificar, predecir consecuencias (ser omnisciente), evaluar alternativas (como en los juegos de ajedrez).
- 6) Conocer los límites de sus propias habilidades y conocimientos.
- 7) Poder distinguir a pesar de la similitud de las situaciones.
- 8) Poder ser original, creando incluso nuevos conceptos o ideas, y hasta utilizando analogías.
- 9) Poder generalizar.
- 10) Poder percibir y modelar el mundo exterior.
- 11) Poder entender y utilizar el lenguaje y sus símbolos.
- 12) Poder medir el desempeño de sus actos, realizando acciones que, se espera obtendrán el mejor desempeño con respecto a las acciones tomadas anteriormente de las cuales se tiene conocimiento de los resultados obtenidos. (c. p. Bargiela, 2007, p 12).

#### **5.2.4. *Lógica Difusa***

Dada la situación de la estimación de proyectos, autores han recurrido a la lógica difusa para su solución. “La lógica de conjuntos difusos o borrosos, como su nombre lo indica, trabaja con conjuntos que no tienen límites perfectamente definidos, es decir, la transición entre la pertenencia y no-pertenencia de una variable a un conjunto es gradual. Se caracteriza por las funciones de pertenencia, que dan flexibilidad a la modelación utilizando expresiones lingüísticas, tales como mucho, poco, leve, severo, escaso, suficiente, caliente, frío, joven, viejo, etc. Surgió la necesidad de solucionar problemas complejos con información imprecisa, para los cuales la matemática y lógica tradicionales no son suficientes. La lógica difusa es un lenguaje que permite trasladar sentencias sofisticadas del lenguaje natural a un formalismo matemático.” (Cardona, 2001, p. 294)

“La lógica difusa fue desarrollada a partir de 1960 por Lotfi Zadeh, guiado por el principio de que las matemáticas pueden ser usadas para encadenar el lenguaje con la inteligencia humana. Algunos conceptos pueden ser mejor definidos con palabras, los conjuntos difusos ayudan a construir mejores modelos de la realidad” (Cardona, 2001, p. 294)

El ingeniero Lotfy Zadeh se dio cuenta de lo que él llamó principio de incompatibilidad: “Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes” (c.p. Pérez, 2005, p. 35).

#### **5.2.5. *Redes Neuronales Artificiales***

##### **5.2.5.1. Origen:**

“Al inicio de la I.A., el conocimiento se representaba usando reglas, redes semánticas, modelos probabilísticos, etc. La metodología para la resolución de problemas era el uso del proceso humano de razonamiento lógico relacionando las causas del problema con los conocimientos que se poseían y luego se procesaba la información secuencialmente.

Al avanzar la I.A., aparecieron problemas complejos en los que no era posible representar el conocimiento de manera explícita ni procesarlo utilizando el razonamiento lógico.

La posible solución a este inconveniente era utilizar estructuras computacionales paralelas basadas en redes neuronales biológicas; así surgen las redes neuronales artificiales” (Bargiela, 2007, p. 11)

### 5.2.5.2. Neurona Artificial:

“Es sabido que el cerebro está compuesto por células especiales llamadas neuronas las cuales se comunican entre sí (se interconectan) a través del axón, esta conexión es llamada sinapsis. La neurona natural contiene un potencial interno continuo llamado potencial de membrana. Cuando este potencial supera un cierto valor umbral, la neurona está en condiciones de transmitir su potencial a la siguiente neurona en la red mediante el axón” (Bargiela, 2003, p. 11)

En la Figura 1, las entradas de la neurona están representadas por las variables  $e_i$ , mientras que los pesos sinápticos están representados por las constantes  $w_{j,i}$ . Cada entrada es ponderada por su respectivo peso sináptico.

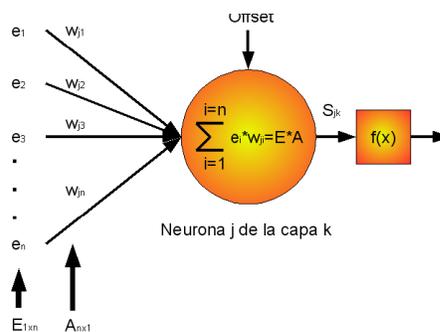


Figura 1. Neurona Artificial (Quiros, 2006)

Los valores obtenidos son sumados entre sí y con el valor del offset, obteniéndose  $S_{j,k}$ . Este resultado corresponde al argumento de la función de salida  $f(S_{j,k})$ , también llamada función de activación. Esta función amplía el campo de problemas que pueden resolver las redes, como

problemas no lineales. Existen varios tipos de funciones de salida. Entre las más conocidas están la función lineal, tangente hiperbólica, función escalón, función saturación, etc.

### **5.2.5.3.Redes neuronales artificiales:**

Una RNA es simplemente una agrupación de neuronas artificiales que trabajan en conjunto para realizar una tarea determinada.

Izaurieta y Saavedra (s/f) presentan las siguientes características de las RNAs (p. 1):

- ◆ Inclínación natural a adquirir el conocimiento a través de la experiencia, el cual es almacenado, al igual que en el cerebro, en el peso relativo de las conexiones inter-neuronales.
- ◆ Tienen una altísima plasticidad y gran adaptabilidad, son capaces de cambiar dinámicamente junto con el medio.
- ◆ Poseen un alto nivel de tolerancia o fallas, es decir, pueden sufrir un daño considerable y continuar teniendo un buen comportamiento, al igual como ocurre en los sistemas biológicos.
- ◆ Tener un comportamiento altamente no-lineal, lo que les permite procesar información procedente de otros fenómenos no-lineales.

### **5.2.5.4.Arquitectura de una RNA:**

Si una neurona se considera como estructura elemental para realizar agrupaciones de ellas, las posibilidades de interconexiones son incógnitas. Se debe escoger una arquitectura de red adecuada para el problema a resolver (Quiros, 2006).

No existen reglas rígidas para la construcción de redes, aunque existen algunas reglas empíricas para su construcción.

Por ejemplo, para resolver problemas complejos se recomienda aumentar el número de neuronas, pero aumentarlo indefinidamente no mejoraría indefinidamente el desempeño de la red. Hay un rango óptimo de número de neuronas para cada problema. Normalmente las

neuronas son agrupadas en capas. Cada capa puede tener distinta cantidad de neuronas y las neuronas de una capa pueden o no estar conectadas entre ellas. Además, las capas pueden conectarse entre ellas de distintas maneras.

En la Figura 2 se muestra una RNA de tres capas, de tamaños  $p$ ;  $q$ ;  $r$  respectivamente.

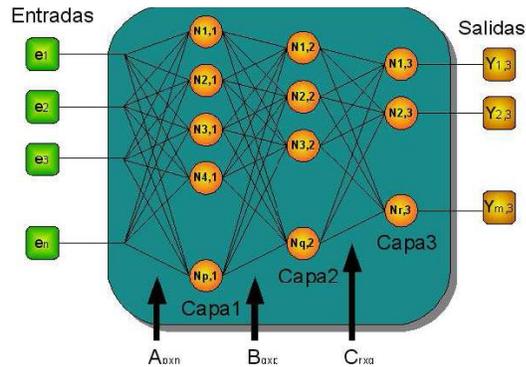


Figura 2. Ejemplo de una RNA (Quiros, 2006)

Como se aprecia en la Figura 2, los pesos entre capas pueden ser representados como matrices de valores reales. Los offsets de las capas pueden ser representados como vectores. Es una manera simple y ordenada de manipular estos valores en un programa computacional. Esta red posee 3 matrices de pesos y 3 vectores de bias (entrada constante de magnitud 1). Estas matrices y vectores representan la "inteligencia" de la RNA.

#### **5.2.5.5. Redes Neuronales Aplicados al Análisis de Datos:**

Los investigadores normalmente, cuando hablan de análisis paramétricos de datos, se centran en la estadística paramétrica. Sin embargo, el campo de las RNAs ha servido para realizar análisis que pueden no alejarse del entendimiento estadístico si se tienen conocimiento de las similitudes de conceptos de los términos y métodos de modelos. A continuación se puede observar dicha similitud (ver Tablas 5 y 6) (Sarle, 1994).

Tabla 5. Equivalencia en la terminología estadística y de redes neuronales. (Sarle, 1994, c.p. Montaña, 2002)

<b>Terminología estadística</b>	<b>Terminología de redes neuronales</b>
Observación	Patrón
Muestra	Datos de entrenamiento
Muestra de validación	Datos de validación, test
Variabes explicativas	Variabes de entrada
Variable de respuesta	Variable de salida
Modelo	Arquitectura
Residual	Error
Error aleatorio	Ruido
Estimación	Entrenamiento, aprendizaje
Interpolación	Generalización
Interacción	Conexión funcional
Coefficientes	Pesos de conexión
Constante	Peso umbral
Regresión y análisis discriminante	Aprendizaje supervisado o heteroasociación
Reducción de datos	Aprendizaje no supervisado o autoasociación
Análisis de cluster	Aprendizaje competitivo

Tabla 6. Equivalencia entre modelos estadísticos y modelos de red neuronal. (Sarle, 1994, c.p. Montaña, 2002)

<b>Modelo estadístico</b>	<b>Modelo de red neuronal</b>
Regresión lineal múltiple	Perceptrón simple con función lineal
Regresión logística	Perceptrón simple con función logística
Función discriminante lineal	Perceptrón simple con función umbral
Regresión no lineal múltiple	Perceptrón multicapa con función lineal en la salida
Función discriminante no lineal	Perceptrón multicapa con función logística en la salida
Análisis de componentes principales	Regla de Oja
Análisis de clústeres	Perceptrón multicapa autoasociativo
K vecinos más cercanos	Mapas autoorganizados de Kohonen
Regresión kernel	Learning Vector Quantization (LVQ)
	Funciones de Base Radial (RBF)

### **5.2.5.6. Análisis del efecto de las variables en una RNA perceptrón multicapa:**

El estudio del efecto o importancia de las variables de entrada en una RNA perceptrón multicapa es uno de los aspectos más críticos en la utilización de las RNAs orientadas al análisis de datos, debido a que el valor de los parámetros obtenidos por la red no tienen una interpretación práctica a diferencia de un modelo de regresión clásico. Como consecuencia, las RNAs se han presentado al usuario como una especie de “caja negra” a partir de las cuales no es posible analizar el papel que desempeña cada variable de entrada en la predicción realizada.

Se ha podido observar que desde finales de los años 80 se han propuesto diferentes métodos dirigidos a interpretar lo aprendido por una red.

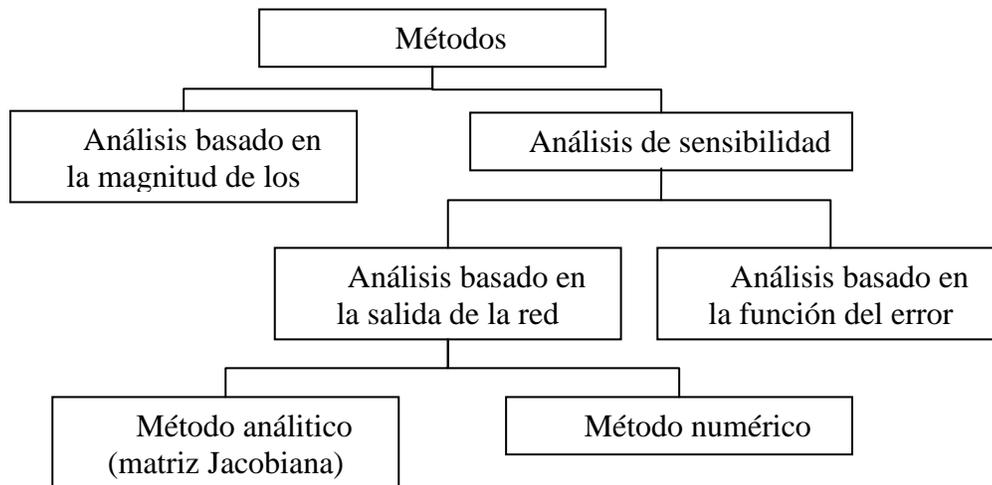


Figura 3. Esquema de los Métodos Interpretativos Propuestos (Montaño, 2002)

Siguiendo el esquema presentado en la Figura 3, estos métodos interpretativos se pueden dividir en dos tipos de metodologías: análisis basado en la magnitud de los pesos y análisis de sensibilidad.

En el trabajo “Redes neuronales artificiales: abriendo la caja negra” (Montaño, Palmer y Fernández, 2002, c.p. Montaño, 2002) hablan del análisis basado en la magnitud de los pesos o

los valores almacenados en la matriz estática de pesos con el propósito de determinar la influencia relativa de cada variable de entrada sobre cada una de las salidas de la red.

Por otro lado, en el trabajo “Numeric sensitivity analysis applied to feedforward neural networks” (Montaño & Palmer, 2002, c.p. Montaño 2002), explican el análisis de sensibilidad usando RNAs basado en la medición del efecto que se observa en una salida o en el error cometido debido al cambio que se produce en una entrada.

El análisis de sensibilidad aplicado sobre una función de error se puede realizar de diversas formas. Una modalidad consiste en ir variando el valor de una de las variables de entrada a lo largo de todo su rango mediante la aplicación de pequeños incrementos o perturbaciones, mientras se mantienen los valores originales de las demás variables de entrada; con esto se busca aislar el comportamiento de la variable en estudio del resto. También se puede optar por restringir la entrada de interés a un valor fijo, eliminando directamente esa entrada. En cualquiera de estos casos, el procedimiento se basa en concluir que la entrada es importante en la predicción realizada por el modelo, si el error aumenta sensiblemente ante el cambio provocado.

El método más habitual de realizar el análisis de sensibilidad consiste en estudiar el efecto de las entradas directamente en la salida estimada por la red. Esto se puede realizar de forma similar al análisis aplicado sobre una función de error, fijando el valor de todas las variables de entrada a su valor medio, excepto una variable sobre la cual se añade ruido o pequeños incrementos. Posteriormente se miden los cambios producidos en la salida de la red (Palmer, Montaño y Calafat, 2000, c.p. Montaño, 2002).

Un procedimiento que goza de una mejor fundamentación matemática se basa en la obtención de la matriz Jacobiana mediante el cálculo de las derivadas parciales de las salidas con respecto a las entradas, esto es,  $\partial Y_k / \partial X_i$ , el cual constituye la versión analítica del análisis de sensibilidad o método ASA (Analytic Sensitivity Analysis) (Montaño, Palmer y Fernández, 2002, c.p. Montaño, 2002).

## **6. METODOLOGIA DE SUPERFICIE DE RESPUESTA**

La metodología de superficie de respuesta (RSM de sus siglas en inglés) es una colección de técnicas estadísticas y matemáticas útiles para el desarrollo, mejoramiento y optimización de procesos (Raymond y Montgomery, 2002 c.p. Carley, Kamneva & Reminga, 2004).

Kamneva y Reminga (2004) expresan que los usos más comunes de la RSM son en aquellas situaciones particulares donde diferentes variables de entrada influyen potencialmente en las características de medida o calidad del proceso, denominado “respuesta”.

El campo de la RSM consiste en la estrategia experimental de explorar el espacio de un proceso, o variables independientes mediante un modelado empírico estadístico para desarrollar una relación aproximada entre la “respuesta” y las variables del Proceso, y lograr métodos de optimización para encontrar los valores de las variables del proceso que producen valores deseables de respuesta.

Según lo expresado anteriormente, se hace tentativa la utilización de este método como técnica para analizar la relación de las variables de interés, una vez que la RNA haya aprendido sus relaciones (“y así descifrar la caja negra”). En este caso se estaría hablando de un Análisis de Sensibilidad basado en la salida de la red mediante este método numérico.

## **7. TIEMPO DE DESARROLLO Y LA EXPERIENCIA**

La experiencia del personal de proyecto tiene un efecto considerable en la productividad. A medida que estos van ganando experiencia desarrollando sus actividades de manera repetitiva, van también ganando la habilidad de desarrollarlo en menos tiempo. Así lo confirma la investigación realizada por Bayona, Calvo, Cuevas & San Feliu (2007) en su trabajo con 22 equipos de 5 a 6 programadores de nivel universitario. Estos 22 equipos se hicieron pasar por

2 ciclos de desarrollo, en el primero se asume su falta de experiencia, y en el segundo ciclo, es utilizado para ver cómo la experiencia del primer ciclo afecta su productividad.

Los resultados a los que llegaron fueron que, a medida que aumenta la experiencia, disminuyen considerablemente los errores de estimación, aumenta la calidad del software al presentarse menos errores, y aumenta la velocidad de programación medida en líneas de códigos por hora.

Esta conclusión confirma lo que representa las *curvas de experiencia* (llamadas también curvas de aprendizaje organizacional), muestran la reducción de costes marginales y medios en forma de aumentos acumulados de la producción. Las curvas de aprendizaje ponen de manifiesto la manera en que los costes variables medios (por unidad) varían en función de la experiencia. Las curvas de la experiencia incluyen también los costes fijos y representan los cambios de costes medios cuando se tienen en cuenta todos los factores.

La teoría de curvas de aprendizaje se basa en tres suposiciones:

- 1) El tiempo necesario para completar una tarea o unidad de producto será menor cada vez que se realice la tarea.
- 2) La tasa de disminución del tiempo por unidad será cada vez menor.
- 3) La reducción en tiempo seguirá un patrón previsible.

### **Causas de la curva de experiencia**

Los efectos de la curva de experiencia no responden a una ley natural, de modo que es necesario interpretar sus causas. La reducción de los costes, que es consecuencia de una relación recíproca, no se produce espontáneamente, sus posibilidades se deben conocer y aprovechar. En tal sentido, se puede utilizar el análisis realizado por Lefcovich (2006), dándole un enfoque más orientado al ámbito de los proyectos:

Eficiencia del equipo de trabajo. La repetición de la tarea genera progresivamente una mayor eficiencia. El gasto se reduce y aumenta la productividad. Este proceso se puede impulsar mediante el entrenamiento y los planes de acción en la gestión de personal.

Organización del trabajo. La organización del trabajo se evidencia de la siguiente manera: a medida que crece el tamaño del proyecto, aumenta el nivel de especialización. De no ser así, hay que reestructurar la organización del equipo de proyecto.

Uniformidad de las tareas. Las ventajas de la curva de experiencia no se podrían aprovechar plenamente sin una uniformidad en las tareas. Sin embargo, la uniformidad conduce a una peligrosa falta de flexibilidad, lo que suele detener la innovación en la manera de gerenciar y desarrollar los proyectos.

La especialización técnica. A medida que aumenta el tamaño del proyecto, se requieren nuevos equipos especializados, lo cual trae consigo una producción más eficiente y con ello más bajos costos.

Modificaciones en el diseño. A medida que se acumula experiencia, las personas pueden tener mayor conocimiento de la relación entre costo y rendimiento. Los productos, tareas, métodos y herramientas del proyecto se pueden modificar para ahorrar material, energía y mano de obra, manteniendo o aumentando su rendimiento.

## **CAPÍTULO III.**

### **MARCO METODOLÓGICO**

En este capítulo se describe la metodología utilizada para la realización de la presente investigación, así como también los lineamientos y las características que posee.

#### **1. TIPO DE INVESTIGACIÓN**

El presente estudio es:

- No experimental: porque los datos obtenidos de la base de datos histórica serán procesados por la RNA sin ninguna manipulación por parte del investigador.
- Explicativo: porque se centra en explicar cómo el tamaño del software medido con la herramienta de puntos de función y la experiencia del equipo de proyecto y su gerente, afectan sobre el esfuerzo necesario para el desarrollo, medido en número de horas/hombres.
- Transversal: porque el estudio no pretende indagar sobre el comportamiento de las variables en el tiempo, sino sobre la relación de las variables para un momento único.

#### **2. UNIDAD DE ANÁLISIS, POBLACIÓN Y MUESTRA**

##### **2.1.Unidad de Análisis**

La unidad de análisis a considerar en este estudio constituye cada proyecto perteneciente de la base de datos Desharnais.

## 2.2.Población

La población la constituye todos los proyectos de software desarrollados a nivel mundial, y registrados en bases de datos con información de puntos de función, lenguaje y experiencia del equipo de proyecto y su gerente.

## 2.3.Tamaño de la Muestra

La muestra está representada por los 77 proyectos de la base de datos Desharnais.

## 3. VARIABLES

### 3.1.Variables Independientes

Las variables independientes constituyen el tamaño del software, el lenguaje utilizado y la experiencia del equipo de proyecto. El tamaño del software está medido en Puntos de Función no ajustados tanto de las entidades como de las transacciones, así con el factor de ajuste asociado a la complejidad del proyecto. La experiencia está medida en años y aplica tanto para el equipo de proyecto como para el gerente de proyecto (ver Figura 4).

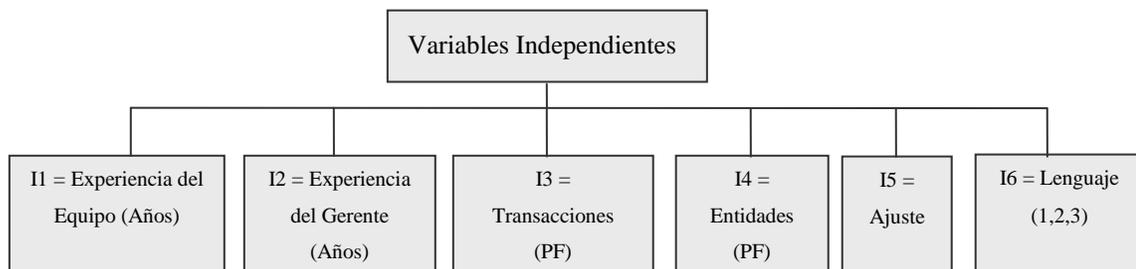


Figura 4. Variables Independientes

### **3.2.Variable Dependiente**

**O = Esfuerzo en Horas/Hombre:** Constituye la sumatoria total del número de horas que tienen que emplear las personas que trabajan para el proyecto.

## **4. ESTRATEGIA DE RECOLECCIÓN Y PROCESAMIENTO Y ANÁLISIS**

### **4.1.Recolección**

La información es adquirida de la base de datos Desharnais, la cual fue construida por J. M. Desharnais (1989) para su trabajo de maestría en la Universidad de Montreal, y posteriormente publicada por “PROMISE Software Engineering”.

### **4.2.Procesamiento**

Obtenida la base de datos, se inicia un proceso de entrenamiento de diferentes RNAs, manteniendo como constante el número de entradas y las salidas, y variando desde 2 el número de nodos en la capa intermedia hasta lograr un Error Relativo Medio (MRE) menor a 5%.

Una vez obtenidas las diferentes RNAs entrenadas (modelos), con la ayuda del software StatGraphic, se pasa a realizar un análisis de sensibilidad numérico (Ver Figura 3) mediante un análisis de superficie de respuesta a cada RNA, lo cual proporciona la R-Cuadrada.

Por otro lado, considerando la variable “Diferencia” expresada en (13), donde  $i$ , representa el  $i$ -ésimo proyecto dentro de la muestra, “R” es el esfuerzo real del proyecto y “E” el valor estimado por la red neuronal.

$$D_i = R_i - E_i \quad (13)$$

Se obtiene la distribución de la variable “D” por cada modelo experimentado. Esta representa una variable aleatoria normal que evalúa la diferencia que se tiene al valor real, por lo tanto, un modelo al ser evaluado en cada caso real que se disponga, debe generar una data con distribución normal, que siempre que su desviación estándar sea “pequeña” alrededor del 0, podemos afirmar que ese modelo está representando la realidad.

Con la ayuda del software StatGraphic, se obtiene un resumen estadístico de dicha variable (ver Tabla 7).

Tabla 7. Resumen estadístico de la variable diferencia.

Mode-lo	Prome-dio	Varian-za	Desvia-ción Estándar	Error Relativo Promedio	Mín.	Máx.	Rango	Sesgo Estandarizado	Intervalo de Confianza 95%	
									Límite Inferior	Limite Superior

De igual manera, mediante la ANOVA simple aplicada a la “Diferencia”, es posible determinar si existen diferencias significativas en la distribución observada para cada modelo.

La R-Cuadrada y el estudio estadístico de la variable “Diferencia” serán indicadores que permiten seleccionar el modelo de RNA más apropiado para el estudio. Los criterios de selección más importantes son la R-Cuadrada y los intervalos de confianza con un 95% de confianza.

### 4.3. Análisis

La idea es utilizar el modelo para probar si las variables que se tomaron en cuenta para definir el esfuerzo así como sus interacciones, son significativas o no, y poder utilizar el modelo para pronosticar el esfuerzo con buenos niveles de confianza para futuros proyectos.

Para ello, se decidió utilizar Análisis de Varianza partiendo de un diseño experimental de superficie de respuesta con 6 variables independientes, y el esfuerzo como variable dependiente.

Se eligió el diseño de Box Behnken, por ser un diseño de superficie de respuesta en un bloque. El número de bloques viene dado por el número de unidades experimentales que bajo unas entradas, responderán de igual manera con su salida. En el presente trabajo, la unidad experimental es el modelo en sí, y su respuesta bajo entradas iguales es siempre la misma; en el caso que se tenga más de una unidad experimental, se hacen varias corridas de experimentos con iguales entradas, para analizar la variación que otorga el cambiar de unidad experimental. Este diseño permite 27 grados de libertad al error en 57 corridas, garantizando poder ver los efectos principales de las variables independientes y sus interacciones.

Los niveles de los factores (variables) que se utilizaron en el diseño vienen dado por la tabla 8:

Tabla 8. Niveles de Valores de los factores utilizados en el diseño.

Variable	Valor		
	Mínimo	Medio	Máximo
Experiencia del Equipo (años)	0	2	4
Experiencia del Gerente (años)	0	3,5	7
Transacciones (PF)	9	335	661
Entidades (PF)	7	197	387
Ajuste	5	28,5	52
Lenguaje	1	2	3

Una vez seleccionada la RNA, mediante el análisis de superficie de respuesta, se observarán gráficas de pareto, curvas y contornos que permiten analizar tanto el efecto principal de cada variable de entrada, como sus interacciones, sobre la variable de esfuerzo o variable respuesta.

Para cerrar, se presentará un ejemplo práctico de la utilización del modelo de estimación del esfuerzo en proyectos de software, haciendo modificaciones en el escenario original de acuerdo a situaciones hipotéticas que ocurren en las organizaciones, para observar de manera práctica cómo se ajusta el esfuerzo a los cambios.

## **CAPÍTULO IV.**

### **PRESENTACIÓN Y ANÁLISIS DE RESULTADOS**

A continuación se presenta el resultado de las diferentes actividades realizadas para el cumplimiento de los objetivos de la investigación. Los pasos se resumen de la siguiente manera:

- 1) Obtención de la base de datos de proyectos.
- 2) Entrenamiento de las redes neuronales con los datos obtenidos.
- 3) Análisis estadísticos de la distribución de la Diferencia (Valor Estimado – Valor Real) por cada modelo.
- 4) Análisis de superficie de cada RNA (obtención de la R-Cuadrada).
- 5) Selección de la RNA más apropiada para el estudio, utilizando la R-Cuadrada y el Rango Mínimo y Máximo de la Diferencia con grado de confianza del 95%.
- 6) Estudio de las gráficas de pareto, de superficie de respuesta y contornos de superficie de respuesta, para determinar el grado de influencia de cada variable de entrada sobre el esfuerzo.
- 7) Aplicación práctica del modelo de estimación de software. Estudio de Casos.

#### **1. SELECCIÓN DE LA ARQUITECTURA DE LA RNA**

Se entrenaron 7 RNAs con diferentes números de nodos en la capa intermedia (de 2 a 8), manteniendo constante el número de variables de entrada, de salida y el método de aprendizaje.

La Tabla 9 que se presenta a continuación muestra los valores que resultaron del análisis de superficie aplicado a cada una de las RNAs.

Tabla 9. Análisis de Superficie de cada modelo estudiado.

Modelo Nro.	Nodos Capa Intermedia	R-cuadrada	R-cuadrada (ajustada por g.l.)	Error estándar del est.	Error absoluto medio	Estadístico Durbin-Watson
1	2	90,7284	81,1002	2244,95	1193,92	2,04221 (P=0,2277)
2	3	88,9859	77,5483	2587,41	1410,36	2,16542 (P=0,3574)
3	4	89,9417	79,4965	2272,0	1223,67	2,68947 (P=0,9052)
4	5	83,7725	66,9208	3009,3	1581,82	2,72556 (P=0,9242)
5	6	82,2191	63,7543	3287,43	1801,98	2,11543 (P=0,3015)
6	7	86,6461	72,7786	3183,62	1761,44	2,46954 (P=0,7225)
7	8	85,9437	71,3468	3209,63	1722,0	2,27914 (P=0,4953)

Según los resultados que se presentan en la Tabla 9, todos los modelos tienen una R-cuadrada mayor a 75%, lo cual significa que las variables independientes en el modelo justifican en gran medida el comportamiento sobre la variable dependiente. Sin embargo, si se considera solo la R-cuadrada como criterio de selección, el modelo Nro. 1 con dos nodos en la capa intermedia se perfila como el de mayor probabilidad de ser elegido, siguiendo en segundo lugar el modelo número 3. Aún así, es importante considerar un estudio más profundo para llegar a una selección con mejores basamentos.

Se obtuvo la variable Diferencia (ver Expresión 12) por cada proyecto ( $i=1 - 76$ ) en cada modelo. Mediante un análisis de varianza simple (ANOVA simple) (ver Tabla 10), con el que se quiere probar la existencia o no de diferencias significativas en la variable Diferencia de acuerdo al modelo, se puede observar que la razón-F es igual a 0,474272. Puesto que el valor-P de la razón-F es mayor que 0,05, no existe una diferencia estadísticamente significativa entre la media de la variable Diferencia entre un Modelo y otro, con un nivel del 95,0% de confianza.

Tabla 10. ANOVA para diferencia por modelo.

Fuente	Suma de Cuadrados	GI	Cuadrado Medio	Razón-F	Valor-P
Entre grupos	1,79065E6	6	298442,	0,47	0,8276
Intra grupos	3,30363E8	525	629263,		
Total (Corr.)	3,32154E8	531			

Al analizar los intervalos de confianza de la variable Diferencia, con un 95% de confianza (ver Tabla 11), se observa que los modelos 1 y 5 no contienen el 0; luego, la hipótesis que dicha diferencia tenga tendencia media a ser 0, no es cierta con un 95% de probabilidad. Por tal motivo se rechazan estos dos modelos.

Tabla 11. Resumen estadístico de la diferencia.

Modelo	Promedio	Varianza	Desviación Estándar	Error Relativo Prom.	Mín.	Máx.	Rango	Sesgo Estandarizado	Intervalo de Confianza 95%	
									Límite Inferior	Límite Superior
1	145,434	2,01109E6	1418,13	35	-3084,0	4662,0	7746,0	0,980721	19,0346	271,834
2	105,908	1,62781E6	1275,85	25	-3340,0	7309,0	10649,0	7,23491	-20,4917	232,307
3	75,1974	277214,	526,511	20	-1140,0	1938,0	3078,0	2,40854	-51,2022	201,597
4	44,5921	245421,	495,4	14	-1166,0	1879,0	3045,0	2,49068	-81,8075	170,992
5	160,395	185498,	430,695	10	-593,0	2215,0	2808,0	9,35296	33,9952	286,794
6	23,7368	45591,0	213,52	6	-460,0	787,0	1247,0	3,57394	-102,663	150,136
7	-7,89474	12223,0	110,558	3	-384,0	375,0	759,0	0,883892	-134,294	118,505

Por otro lado, los modelos número 2 y 4 también pueden ser rechazados porque, aunque el intervalo de confianza contiene el 0, también son los de variabilidad mayor.

Ahora bien, si se observa la similitud entre los rangos del intervalo de confianza, el promedio, la varianza y la R-Cuadrada que tienen los modelos 6 y 7, hace intuir que ambos

modelos guardan cierta similitud, permitiendo descartar alguno de los dos. Para el presente estudio fue seleccionado aquel que tiene mayor precisión, en este caso corresponde al modelo número 7.

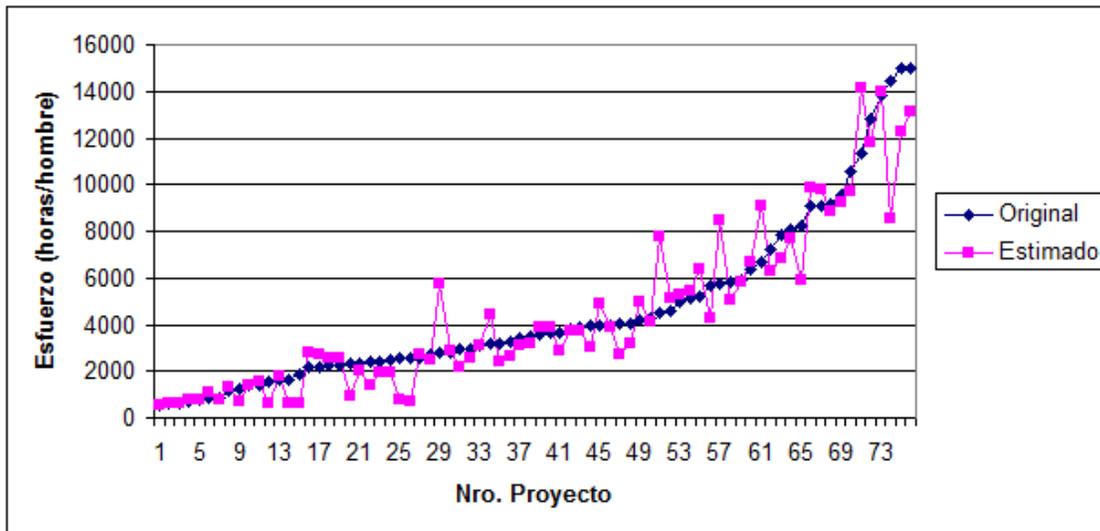
Finalmente quedan los modelos 3 y 7 que, según la importancia dada a la R-Cuadrada o a la Diferencia, se puede seleccionar uno u otro. Para el presente análisis, en vista que se busca experimentar con las variables para ver sus efectos en el resultado, se le dio mayor importancia a la mejor R-Cuadrada, pues su resultado está justificado en mayor porcentaje por las variables que se están estudiando; por tal motivo se seleccionó el modelo número 3. Sin embargo y tal como puede ser visto a continuación, los resultados arrojados en el modelo 7 fueron también considerados en el análisis.

## **2. RESULTADOS DEL ENTRENAMIENTO**

A continuación se presenta gráficamente el valor original del esfuerzo medido en Horas/Hombre de cada proyecto de la muestra, y el valor estimado por cada RNA después de su entrenamiento. Para la experimentación (los entrenamientos) se utilizó un equipo Laptop TOSHIBA Intel® Pentium® M Procesador 1.73 GHz y un 1 Gb de RAM.

### **2.1. Modelo 3**

El modelo número 3 representa la RNA con 4 nodos en la capa intermedia. En la Gráfica 4 se puede observar la diferencia existente entre el valor Original y el Estimado para cada proyecto de la muestra.



Gráfica 4. Original y Estimado del Modelo 3.

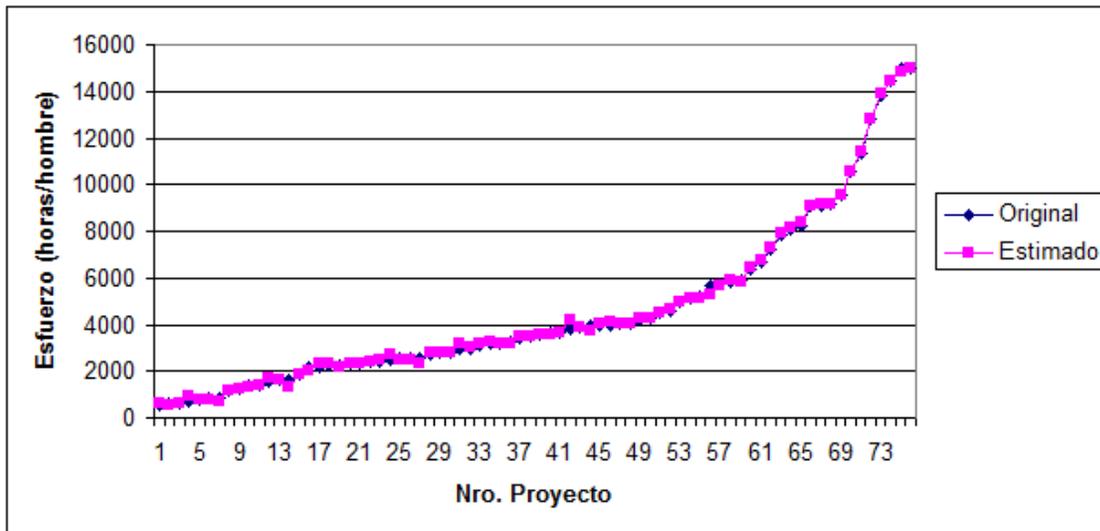
Tiempo que duró el proceso de entrenamiento: 457 Segundos

Ciclos: 2298000

Error cuadrático medio: 0.004080

## 2.2.Modelo 7

El modelo número 7 representa la RNA con 8 nodos en la capa intermedia. En la Gráfica 5 se puede observar la diferencia existente entre el valor Original y el Estimado para cada proyecto de la muestra. A simple vista es posible afirmar que el modelo 7 puede responder con mayor precisión a los proyectos contenidos en la muestra, comparado con el modelo número 3, ya que presenta menor variabilidad alrededor del valor real, pero queda abierta la siguiente pregunta: ¿será posible que pueda predecir con la misma precisión nuevos casos de proyectos?



Gráfica 5. Original y Estimado del Modelo 7.

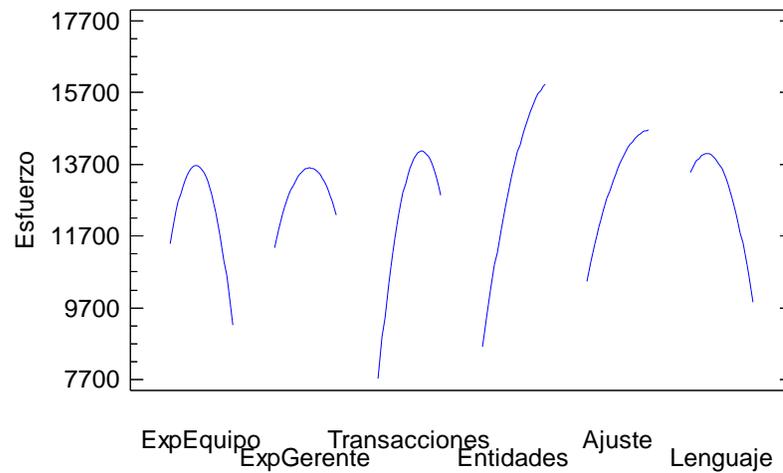
Tiempo que duró el proceso de entrenamiento: 6170 Segundos

Ciclos: 8892000

Error cuadrático medio: 0.000028

### 3. ANÁLISIS DEL ESFUERZO PRONOSTICADO A TRAVÉS DEL MODELO 3

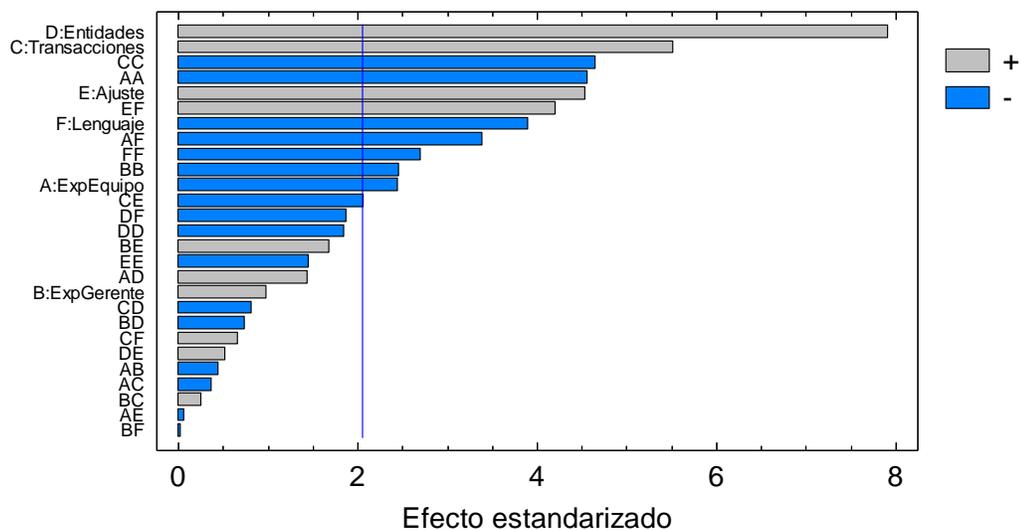
En la Gráfica 6 se presentan los efectos principales de las variables independientes sobre el esfuerzo, cuando estas varían entre sus tres niveles (mínimo, medio y máximo) definidos por el dominio de la data real recabada. Se observa que los efectos principales según este modelo son fuertemente no lineales, y que prácticamente el máximo esfuerzo se tiene en valores medios de las variables independientes, excepto para las entidades y ajuste, donde el esfuerzo máximo se da cuando las entidades o ajuste toman su valor máximo.



Gráfica 6. Efectos Principales para Esfuerzo del Modelo 3.

En particular, si se observa el efecto de la Experiencia del Equipo, cuando esta variable está en su valor mínimo, el esfuerzo es menor a cuando ésta está en niveles medios; sin embargo, el esfuerzo es mínimo al tener el Equipo la máxima experiencia. Luego, el modelo dice que los equipos más experimentados son los que aportan menor esfuerzo pero, no es cierto que al ganar experiencia el esfuerzo siempre disminuye.

En la Gráfica 7 se muestran los niveles de significación de la prueba ANOVA tanto de los efectos principales como la de sus interacciones, así como la línea que divide las zonas de aceptación y rechazo; las variables que superan la línea son las que resultan significativas en la prueba ANOVA; es decir, las variaciones en el Esfuerzo son explicadas en un 95% por las variaciones de estos factores.



Gráfica 7. Pareto estandarizado para Esfuerzo del Modelo 3.

El modelo no lineal obtenido en el análisis de superficie posee coeficientes que se muestran en la Tabla 12.

Tabla 12. Coeficiente de Regresión para Esfuerzo del Modelo 3

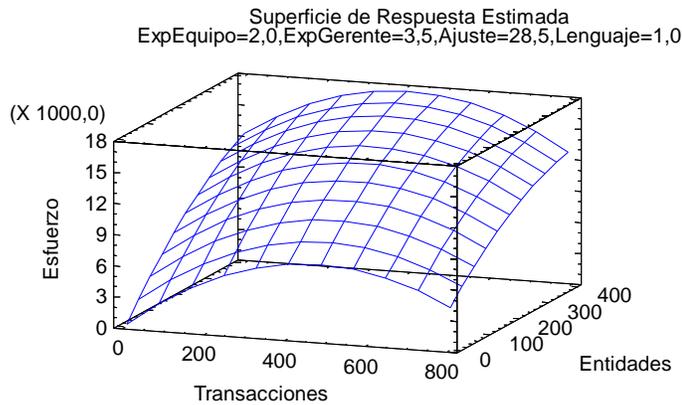
Coefficiente	Estimado	Coefficiente	Estimado
constante	-11139,6	BC	0,175394
A:ExpEquipo	5319,39	BD	-0,890789
B:ExpGerente	1025,43	BE	11,5988
C:Transacciones	34,8598	BF	-5,32143
D:Entidades	49,1678	CC	-0,0310074
E:Ajuste	-76,185	CD	-0,0106272
F:Lenguaje	5660,72	CE	-0,216861
AA	-806,795	CF	1,16123
AB	-51,375	DD	-0,0361165
AC	-0,446894	DE	0,0926372
AD	2,14112	DF	-7,93684
AE	-1,03989	EE	-1,85992
AF	-1361,56	EF	143,75
BB	-142,195	FF	-1912,85

Según el Pareto de la Gráfica 7, las variables principales más significativas en la variación del esfuerzo en proyectos de software son: el alcance del proyecto, representado por las variables Entidades, Transacciones y Ajuste (complejidad); el lenguaje y la Experiencia del Equipo. Las interacciones significativas son: Ajuste - Lenguaje, Experiencia del Equipo - Lenguaje y Transacciones - Ajuste, aunque esta última está ya en el límite entre aceptación y rechazo.

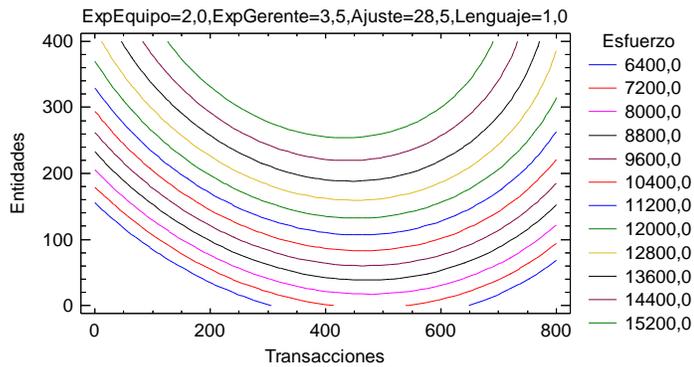
En general, el sentido del efecto (positivo o negativo) no tiene mucho caso que se analice, ya que según lo observado en la Gráfica 6, los efectos no son lineales. Por otro lado, en la Gráfica 7 se observa que existen efectos dobles (Ej: AA, CC), que son significativos; estos efectos tienen sentido en el cálculo, pero no en una interpretación física del modelo; este tipo de interacciones tienen una interpretación real para explicar ciertos modelos sociales que tienen que ver con aprendizaje de individuos, pero aquí solo tienen efecto como variables en el cálculo del pronóstico (ver coeficientes en la Tabla 12).

### **3.1. Esfuerzo en función a las Entidades y las Transacciones**

A continuación, se realiza un análisis del efecto que tienen las entidades y transacciones en conjunto, sobre el esfuerzo necesario para el desarrollo de los proyectos según el modelo número 3. Para el presente análisis, se mantiene la experiencia del gerente, la experiencia del equipo y del ajuste en un valor central, según el rango de la muestra, y se varía el lenguaje entre sus tres posibles valores (1, 2 y 3) para observar su comportamiento según el caso.



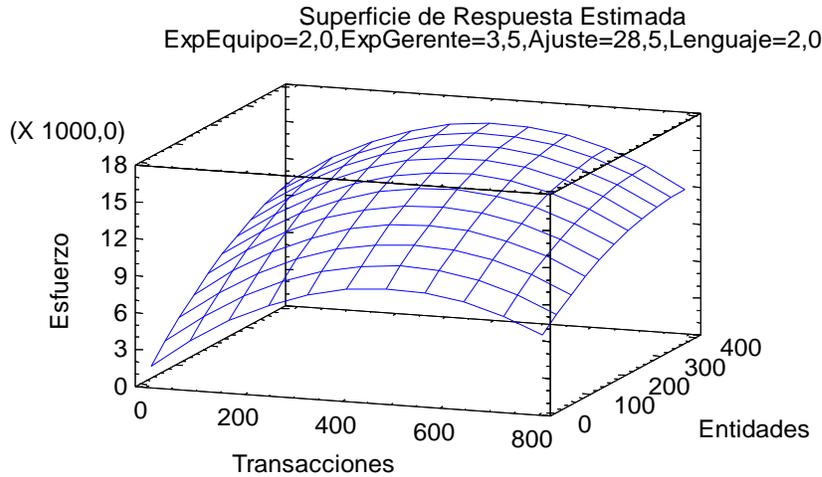
Gráfica 8. Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 1, Modelo 3.



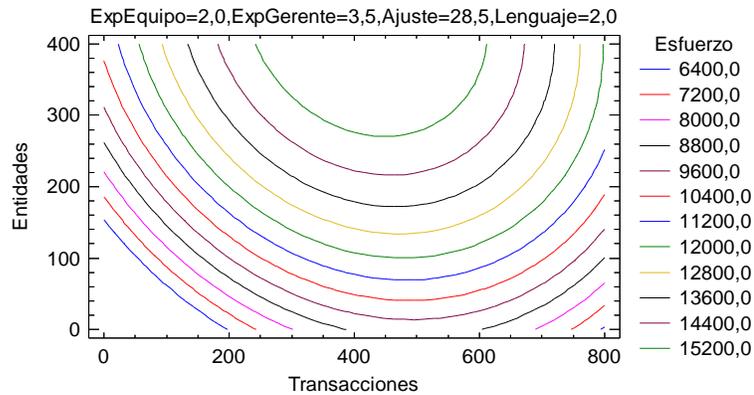
Gráfica 9. Contorno de Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 1, Modelo 3.

En las Gráficas 8 y 9 se puede observar la influencia creciente de las entidades sobre el esfuerzo, igualmente las transacciones; sin embargo, esta última va disminuyendo progresivamente. Según la Gráfica 9, un proyecto con entidades de 300 PF y transacciones de 200 PF puede costar lo mismo que un proyecto con el mismo número de PF en las entidades y transacciones de 600 PF. Aunque suene ilógico, es posible que a medida que se incrementa los PF a nivel de transacciones, se vea más necesaria la reutilización de funciones, bibliotecas predefinidas, programación orientada a objetos, modularización, etc., que permitan la disminución de las horas hombre necesarias. Para poder probar esta hipótesis, sería de gran utilidad contar con el porcentaje de reutilización de código en cada proyecto. Otra hipótesis posible es la falta de datos que representen casos donde las transacciones superen

significativamente a las entidades. En informática, las transacciones y las entidades guardan una relación lógica, por lo que no pueden existir transacciones si no existen los datos.



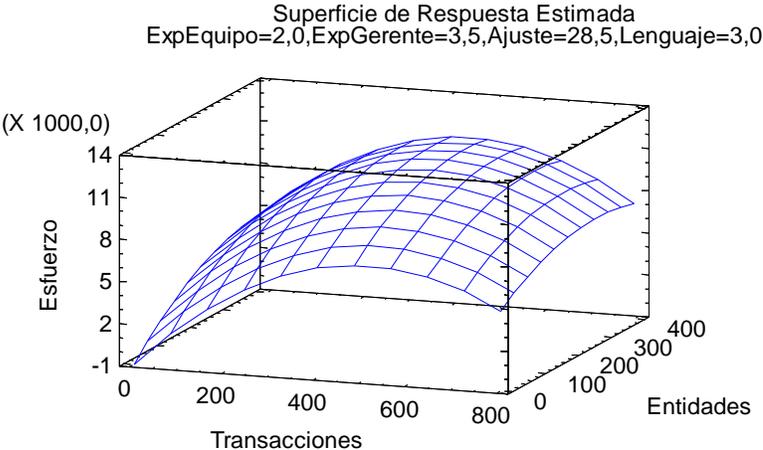
Gráfica 10. Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 2, Modelo 3.



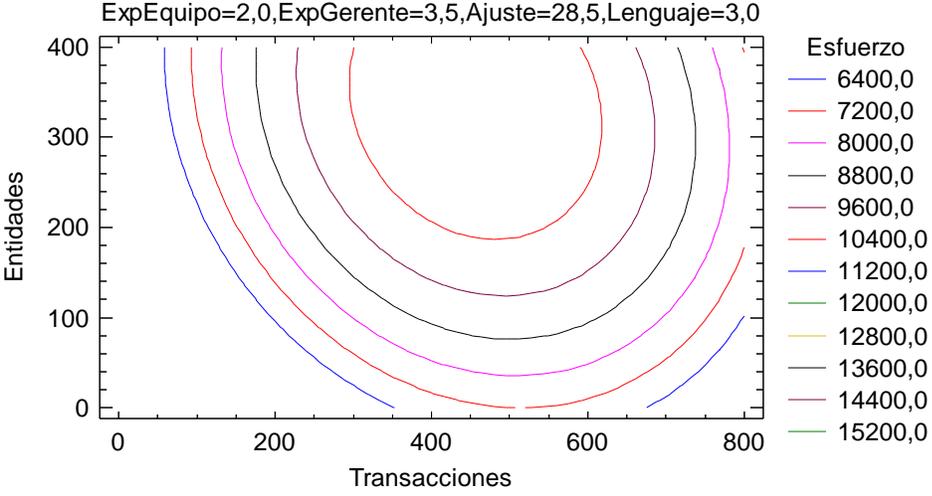
Gráfica 11. Contornos de la Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 2, Modelo 3.

En las Gráficas 10 y 11, cambiando el lenguaje a categoría 2, es posible observar cómo, a medida que se incrementa las entidades y las transacciones, los tiempos son menores en comparación con el lenguaje de categoría 1. En el caso hipotético en que un proyecto X sea posible desarrollarlo con un lenguaje de categoría 1, 2 o 3 indistintamente, manteniendo un equipo de dos años de experiencia y un gerente con 3.5 años de experiencia, un número de

entidades de 300 PF y un número de transacciones de 200 PF, con el lenguaje número 1 necesitaría 14.500 horas/hombre mientras que, utilizando la categoría 2 de lenguaje, 13.600 horas/hombre; ahora bien, al observar la Gráfica 13 (Lenguaje 3), es posible darse cuenta que para la categoría 3 de lenguaje, este valor disminuye a 8.800 horas/hombre, lo que evidencia una disminución cada vez mayor en las horas hombre necesarias para el proyecto.



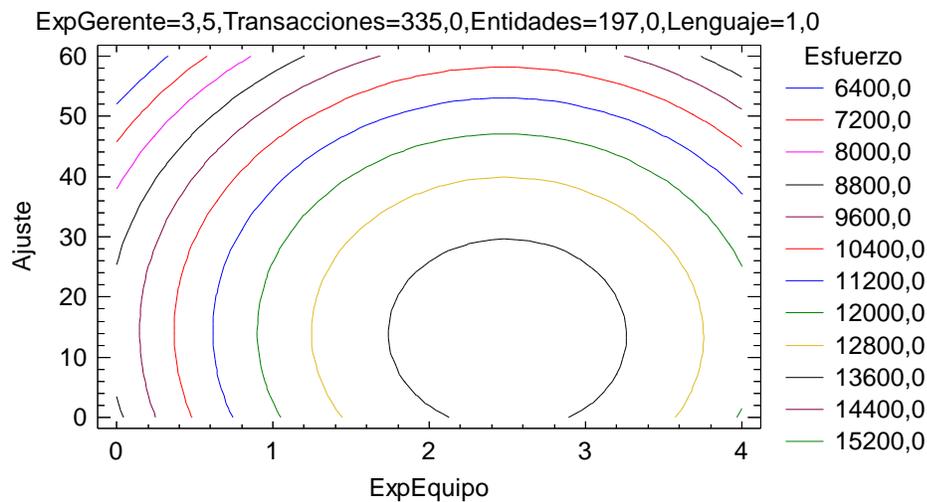
Gráfica 12. Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 3, Modelo 3.



Gráfica 13. Contornos de la Superficie de Respuesta Estimada de las Transacciones y Entidades sobre Esfuerzo para el Lenguaje 3, Modelo 3.

### 3.2.Esfuerzo en función a la Experiencia del Equipo y la Complejidad del Proyecto (ajuste)

A continuación, se realiza un análisis del efecto generado por la experiencia del equipo y la complejidad del proyecto (ajuste), sobre el esfuerzo necesario para el desarrollo del proyecto. Para el presente análisis, la experiencia del gerente, las transacciones y las entidades se mantienen en un valor constante medio, mientras que el lenguaje se varía entre sus valores posibles (1, 2 y 3) para poder observar su efecto.

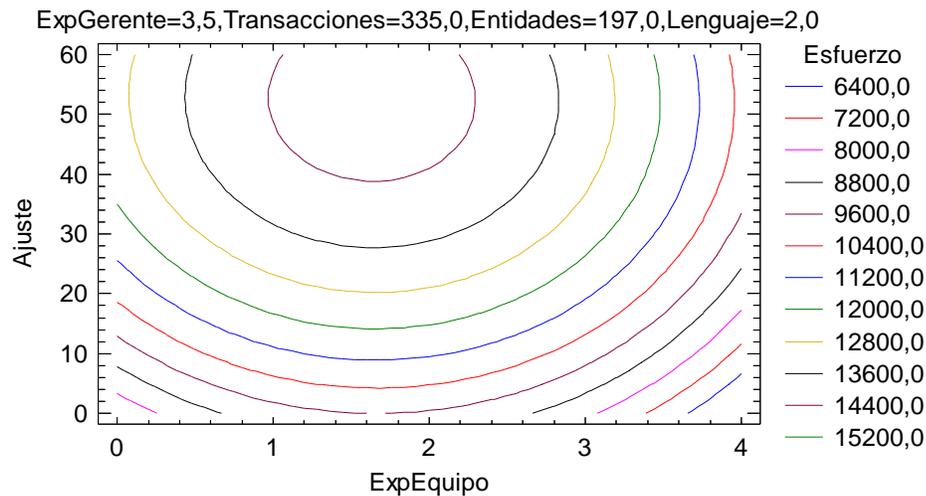


Gráfica 14. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Ajuste sobre el Esfuerzo para el Lenguaje 1, Modelo 3.

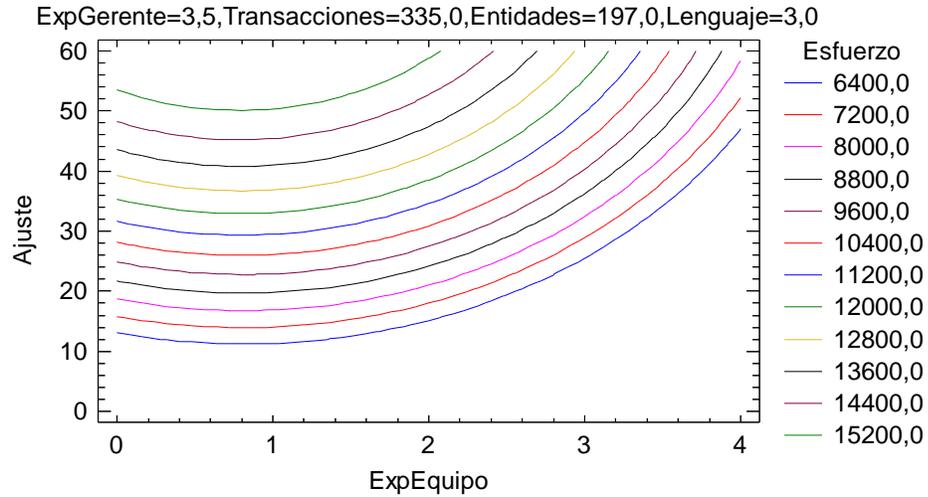
En la Gráfica 14 se puede observar cómo, a medida que aumenta el ajuste, aumenta igualmente el tiempo de desarrollo. Sin embargo, la variable Experiencia del Equipo tiene un comportamiento diferente, los primeros dos años aumenta el tiempo de desarrollo para luego bajar progresivamente. Esto puede que se deba a un proceso de maduración donde empiezan a darse cuenta de todas las actividades implícitas a la complejidad del proyecto, por lo que cada vez se requiere más tiempo para ir mejorando la calidad. Posterior a los 2 años y medios, se afianzan los conocimientos y empieza una disminución progresiva del tiempo necesario para el mismo desarrollo. Para poder comprobar esta hipótesis es necesario contar con una variable

que mida la calidad del software, rotación de personal, nivel de conflictividad grupal y número de personas en el equipo de proyecto.

Analizando la Gráfica 15 con un lenguaje de categoría 2, se observa como este proceso hipotético de comprensión de la complejidad de los proyectos de software ocurre antes de los dos años. Y en el caso de la categoría tres (ver Gráfica 16), este proceso se reduce a menos de un año. Esto puede significar que los equipos, mediante lenguajes de modelado con mayor nivel de abstracción, pueden atacar la complejidad del problema con más facilidad. Según la data observada, la experiencia del equipo es un factor de gran importancia para la reducción de las horas hombres frente a proyectos de diferentes niveles de complejidad.



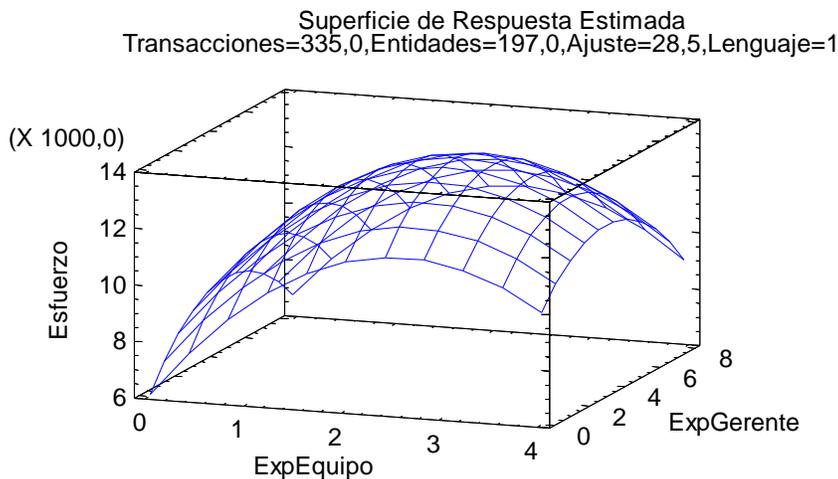
Gráfica 15. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Ajuste sobre el Esfuerzo para el Lenguaje 2, Modelo 3.



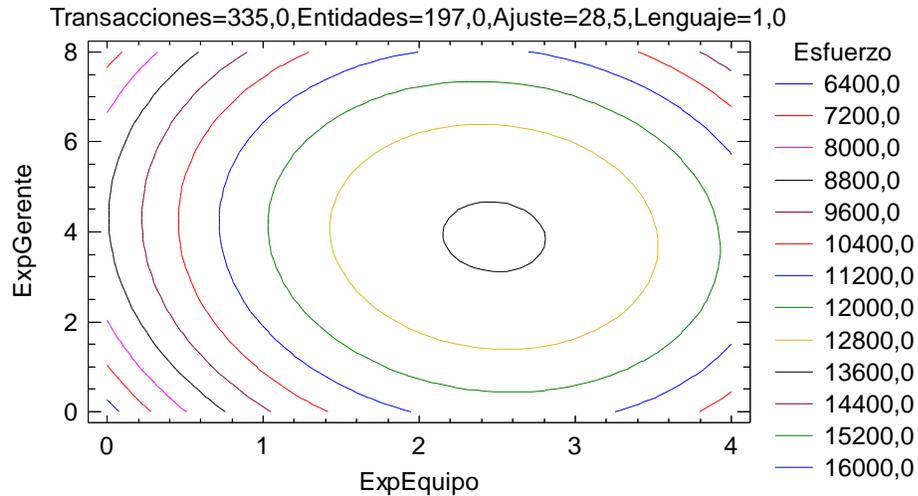
Gráfica 16. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Ajuste sobre el Esfuerzo para el Lenguaje 3, Modelo 3.

### 3.3.Esfuerzo en función a la Experiencia del Gerente y Experiencia del Equipo

A continuación, se realiza el análisis de las variables Experiencia del Gerente y Experiencia del Equipo sobre el esfuerzo necesario para el desarrollo de un proyecto. Al igual que los análisis anteriores, el resto de las variables se mantienen en un valor constante, mientras que el lenguaje se fija en cada uno de sus valores posibles.

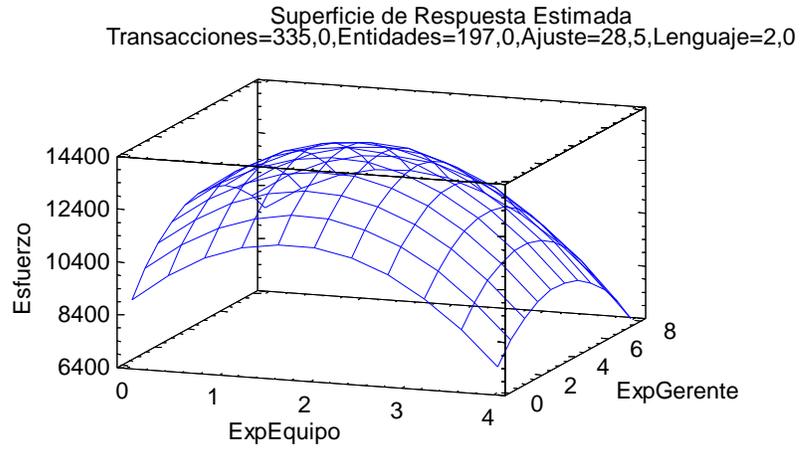


Gráfica 17. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 1, Modelo 3.

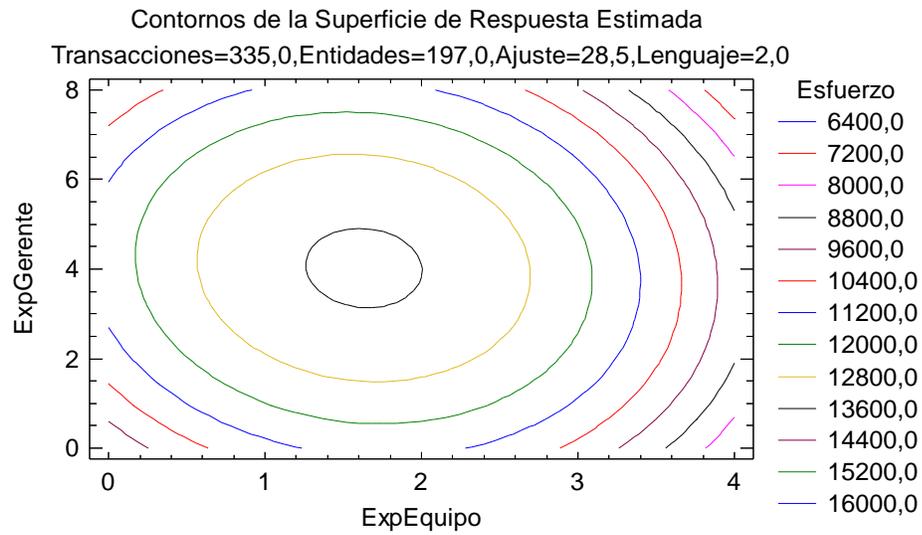


Gráfica 18. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 1, Modelo 3.

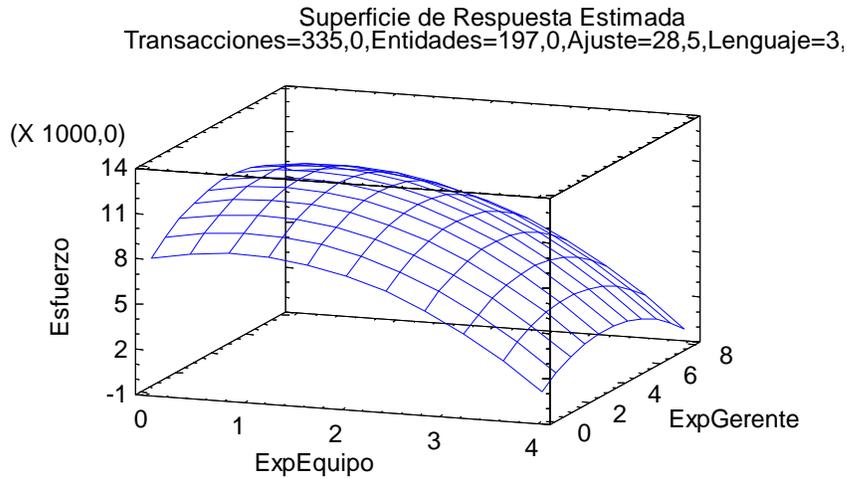
Al variar la experiencia del Gerente entre el valor mínimo de 0 años hasta el valor máximo de 7 años, se observa un efecto positivo en el esfuerzo, que va disminuyendo progresivamente hasta los 3 años, donde empieza a ser negativo (ver Gráficas 17 y 18). El aumento del esfuerzo en los primeros años pudiera estar relacionado con el grado en que el gerente del proyecto va comprendiendo y aplicando las diferentes áreas de conocimiento de la gerencia de proyectos, lo que implica un aumento en el número de horas necesarias. Posteriormente, se inicia una disminución en el esfuerzo necesario para el desarrollo del proyecto, que pudiera sugerir el dominio y mejoramiento de las habilidades necesarias para la gerencia de proyecto. Para poder justificar esta hipótesis, nuevamente se requiere poder disponer de una variable que mida la calidad del software, a su vez, esto pudiera llevar a estudiar la posibilidad de que los gerentes de la muestra adquirieron sus conocimientos de gerencia empíricamente.



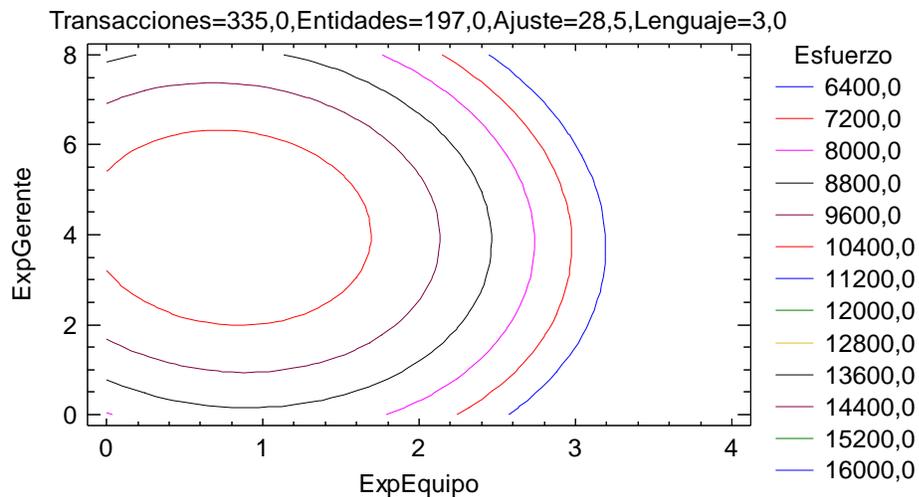
Gráfica 19. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 2, Modelo 3.



Gráfica 20. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 2, Modelo 3.



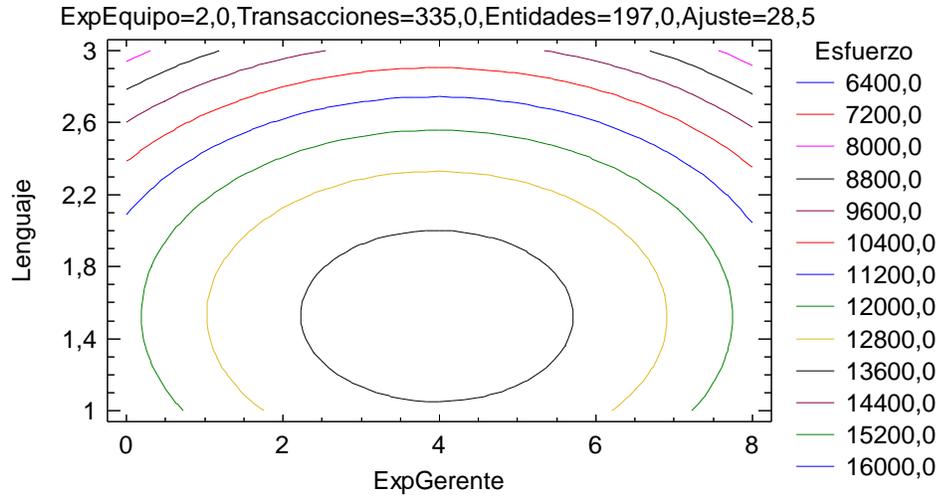
Gráfica 21. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 3, Modelo 3.



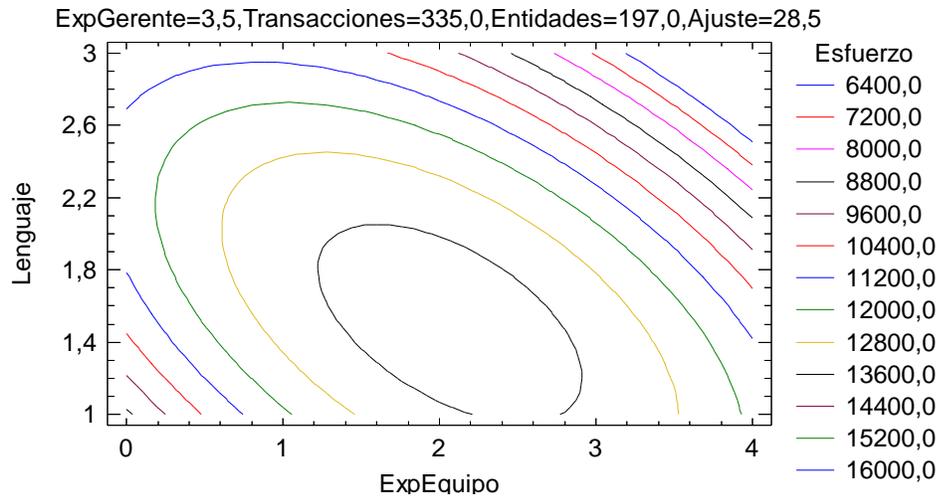
Gráfica 22. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 3, Modelo 3.

En la Gráfica 22 con lenguaje de categoría 3, se puede observar cómo los proyectos con esfuerzos de 12800 y 13600 horas/hombre desaparecen por completo. Es de notar cómo la herramienta de desarrollo influye profundamente en el equipo desarrollador, y casi nada sobre el gerente, lo que tiene una justificación razonable si se piensa que el lenguaje de programación no es una herramienta del gerente de proyectos. Aunque la variable Categoría

del Lenguaje no es continua, se supondrá su continuidad para poder visualizar su efecto en las siguientes gráficas (Gráfica 23 y 24).



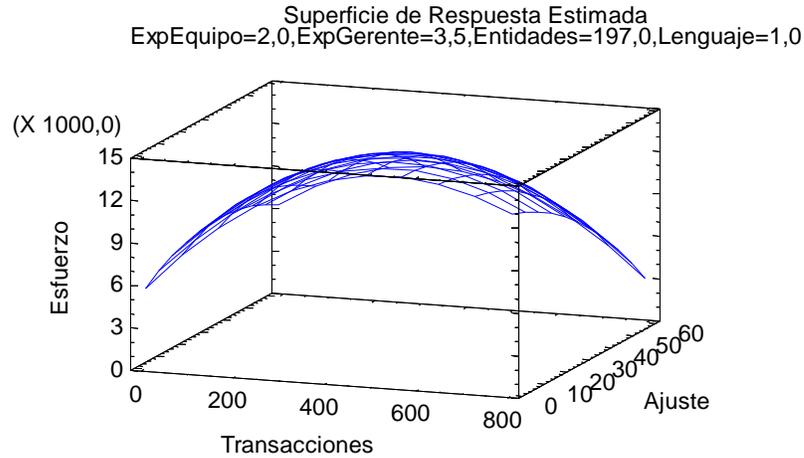
Gráfica 23. Contornos de la Superficie de Respuesta de la Experiencia del Gerente y el Lenguaje sobre el Esfuerzo, Modelo 3.



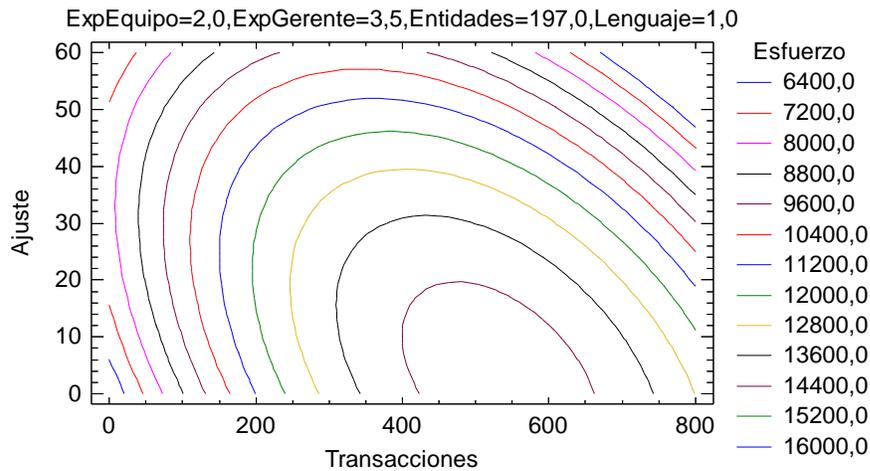
Gráfica 24. Contornos de la Superficie de Respuesta de la Experiencia del Equipo y el Lenguaje sobre el Esfuerzo, Modelo 3.

### 3.1.Esfuerzo en función a las Transacciones y el Ajuste

Existe un significativo efecto combinado de las transacciones y el ajuste sobre el esfuerzo, tal como se observa en la Gráfica de Pareto 7. Por tal motivo, a continuación se realiza un análisis de la misma.



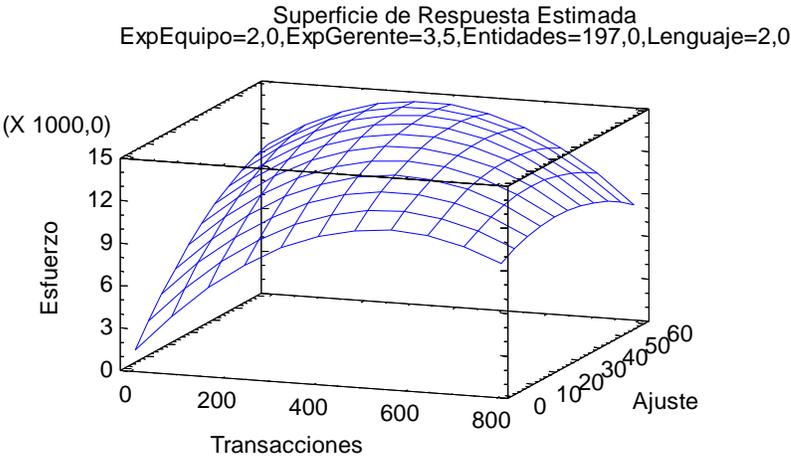
Gráfica 25. Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 1, Modelo 3.



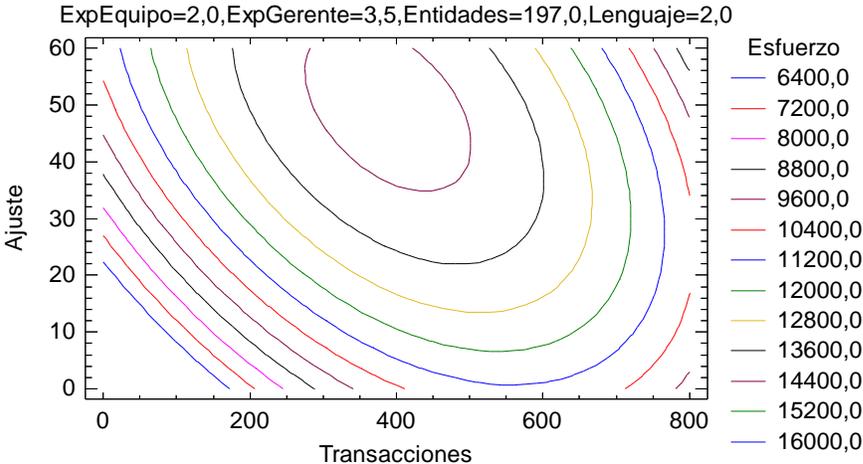
Gráfica 26. Contornos de la Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 1, Modelo 3.

En las Gráficas 25 y 26, es posible observar cómo las transacciones afectan de manera diferente el esfuerzo, a medida que el ajuste aumenta o disminuye para el lenguaje de

categoría 1. El ajuste crea un efecto no lineal en la relación entre las transacciones y el esfuerzo, haciendo incluso que en ciertas circunstancias, cuando la complejidad es alta, a mayores transacciones, el esfuerzo disminuye. Ya este comportamiento se había observado en las Gráfica 8 y 9 con su correspondiente hipótesis. En vista de que las entidades se mantienen constantes en su valor medio de 197 PF, y las transacciones se mueven entre su valor mínimo hasta su valor máximo, es posible que la gráfica esté rompiendo con la relación coherente que existe entre las Transacciones y las Entidades, tal como se expresó en el punto 4 referido a la Base de Datos Histórica de Proyectos.

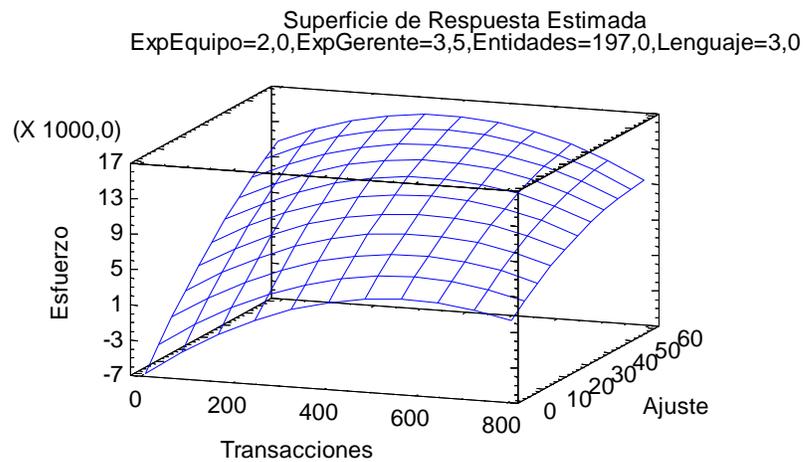


Gráfica 27. Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 2, Modelo 3.

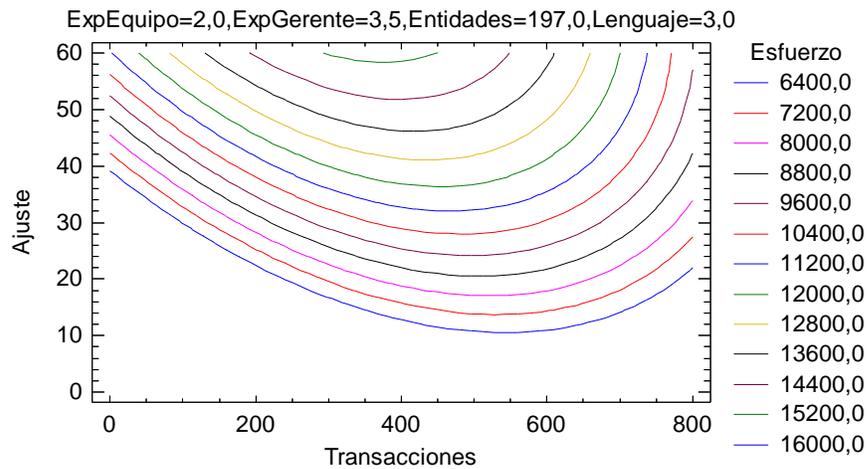


Gráfica 28. Contornos de la Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 2, Modelo 3.

En las Gráficas 27 y 28, donde se aplica la categoría de lenguaje 2, se puede observar el mismo efecto negativo de las transacciones cuando el ajuste toma sus valores más altos, sin embargo esto ocurre en menos proporción.



Gráfica 29. Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 3, Modelo 3.

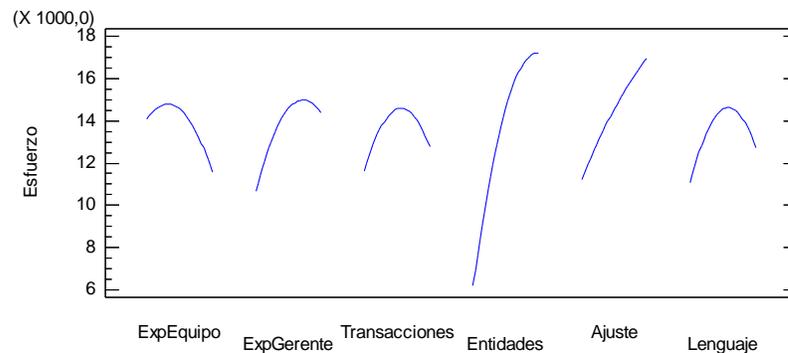


Gráfica 30. Contornos de la Superficie de Respuesta de las Transacciones y el Ajuste sobre el Esfuerzo para el Lenguaje 3, Modelo 3.

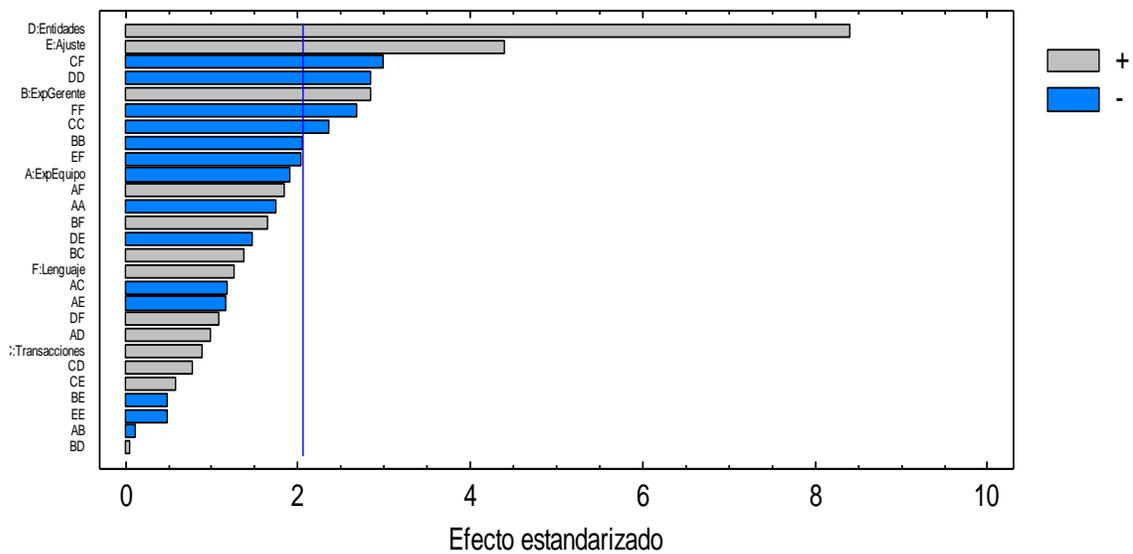
Finalmente, observando las mismas gráficas para el lenguaje de categoría 3 (Gráficas 29 y 30), se puede observar cómo el efecto negativo de las transacciones se ve aun más reducido cuando el ajuste aumenta. Esto nos permite afirmar que para todos los lenguajes, el efecto de las transacciones sobre el esfuerzo es evidentemente positivo cuando el ajuste se mantiene en valores bajos, pero a medida que este aumenta, se hace más confuso el efecto que pueda generar el número de PF asociados a las transacciones del proyecto. La complejidad (medida con la variable “Ajuste”), hace más difícil tener una relación lineal entre las transacciones y el esfuerzo.

#### 4. ANÁLISIS DEL ESFUERZO PRONOSTICADO A TRAVÉS DEL MODELO 7

Siguiendo el mismo procedimiento aplicado en el Modelo número 3, se logró realizar un análisis de superficie de segundo nivel del modelo número 7 que permite determinar la importancia de cada una de las variables en estudio. En la Gráfica 31 se presentan los efectos principales de las variables independientes sobre el esfuerzo, cuando estas varían entre sus tres niveles (mínimo, medio y máximo) definidos por el dominio de la data real recabada.



Gráfica 31. Efectos Principales para Esfuerzo del Modelo 7.



Gráfica 32. Diagrama de Pareto Estandarizado para Esfuerzo del Modelo 7.

El modelo no lineal obtenido en el análisis de superficie posee los coeficientes que se muestran en la Tabla 13.

Tabla 13. Coeficiente de Regresión para Esfuerzo del Modelo 7.

Coefficiente	Estimado	Coefficiente	Estimado
Constante	-19852,2	BC	1,37303
A:ExpEquipo	176,295	BD	0,0881579
B:ExpGerente	334,412	BE	-4,82295
C:Transacciones	25,4678	BF	536,786
D:Entidades	48,573	CC	-0,0222615
E:Ajuste	488,004	CD	0,0143183
F:Lenguaje	11609,3	CE	0,0884186
AA	-438,694	CF	-7,37117
AB	-18,4464	DD	-0,079239
AC	-2,07266	DE	-0,37738
AD	2,09227	DF	6,55592
AE	-28,1915	EE	-0,878578
AF	1050,19	EF	-98,617
BB	-168,026	FF	-2691,49

En este modelo mejor entrenado con los datos obtenidos en la muestra (ver Gráfica 5), las entidades siguen siendo un factor para incrementar el esfuerzo; sin embargo, existen grandes diferencias con respecto al modelo número 3 estudiado anteriormente (ver Gráficas 31 y 32). A continuación se enumeran los aspectos más notables:

- 1) El número de PF de las Transacciones, aunque sigue siendo un factor significativo, ya no tiene el mismo rango de variación que el modelo anterior.
- 2) Aunque la experiencia gerencial continua siendo un factor significativo para el esfuerzo de desarrollo, su papel se invierte, es decir, la experiencia del gerente ayuda a aumentar el esfuerzo, aunque se puede evidenciar que a medida que aumenta la experiencia, en cierto punto ésta empieza a disminuir el esfuerzo, sin embargo, no lo hace en tanto grado como ocurre en el modelo 3.
- 3) La experiencia del equipo deja de ser un factor significativo en el modelo, aunque continua siendo una variable que permite disminuir el esfuerzo.

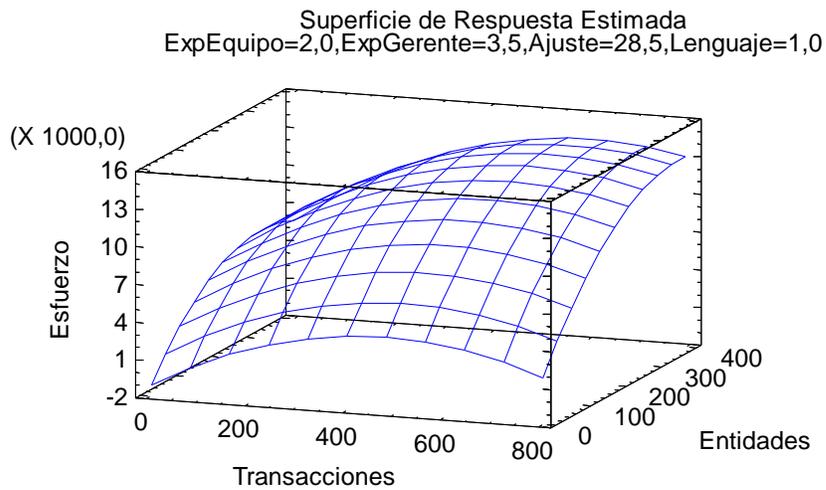
Por tal motivo, a continuación se presenta un análisis más detallado de la influencia de las variables transacciones, y experiencias del equipo de proyecto y del gerente sobre el esfuerzo necesario para el desarrollo del proyecto.

#### **4.1.Esfuerzo en función a las entidades y las transacciones**

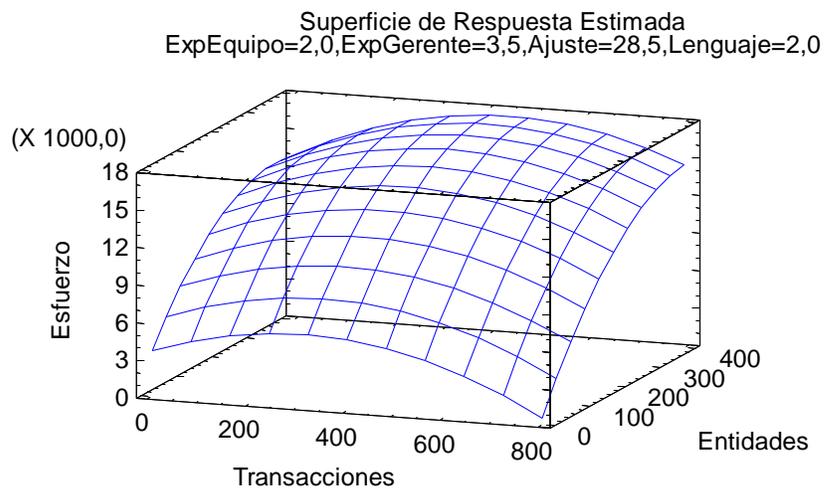
A continuación se realiza un análisis del efecto causado por las entidades y las transacciones sobre el esfuerzo, a medida que éstas varían dentro del dominio de la muestra. Al igual que en los análisis anteriores, se fijan el resto de las variables en valores constantes, exceptuando el lenguaje, el cual se estudia por cada valor posible.

Si se observan las gráficas de superficie de respuesta, Gráficas 34 y 35, es posible ver cómo los lenguajes de categoría 2 y 3 causan un efecto poco creíble en la relación existente entre las transacciones y el esfuerzo. A mayor transacciones, menor esfuerzo. Este comportamiento ya

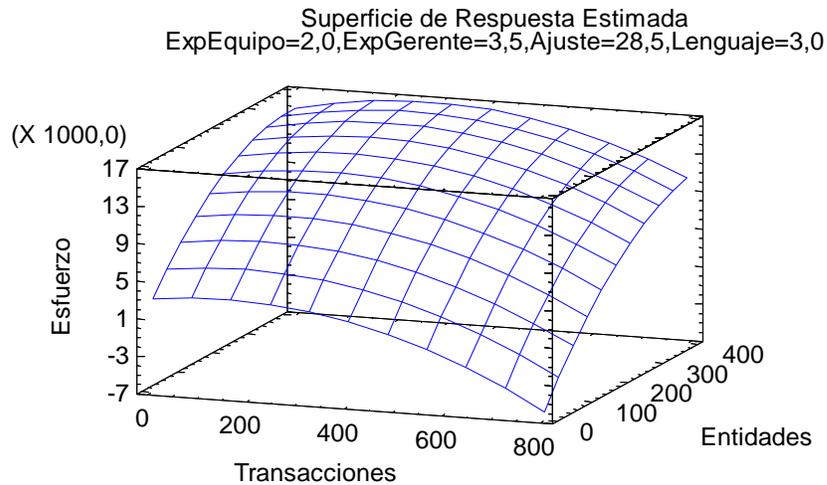
es un indicio fuerte para no considerar consistente el modelo número 7. Un comportamiento similar, pero en menos proporciones se había observado en el modelo 3, cuando se estudió la interacción transacciones – ajuste sobre el esfuerzo.



Gráfica 33. Superficie de Respuesta de las Transacciones y Entidades sobre el Esfuerzo para el Lenguaje 1, Modelo 7.



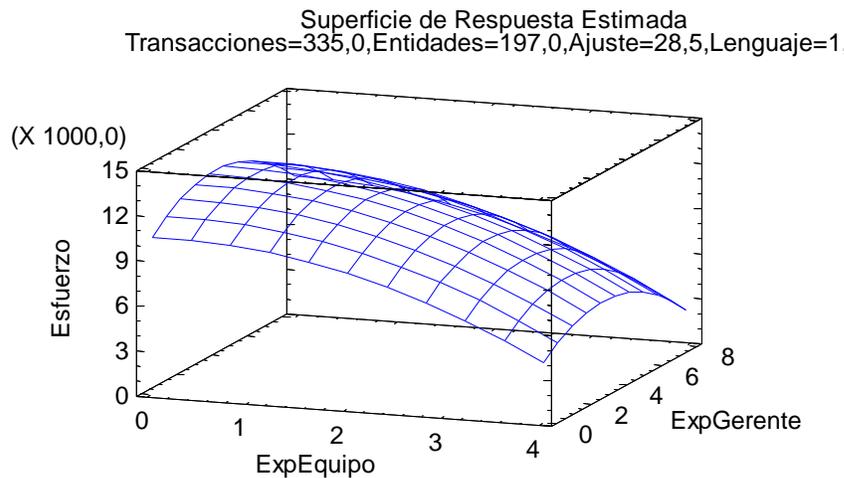
Gráfica 34. Superficie de Respuesta de las Transacciones y Entidades sobre el Esfuerzo para el Lenguaje 2, Modelo 7.



Gráfica 35. Superficie de Respuesta de las Transacciones y Entidades sobre el Esfuerzo para el Lenguaje 3, Modelo 7.

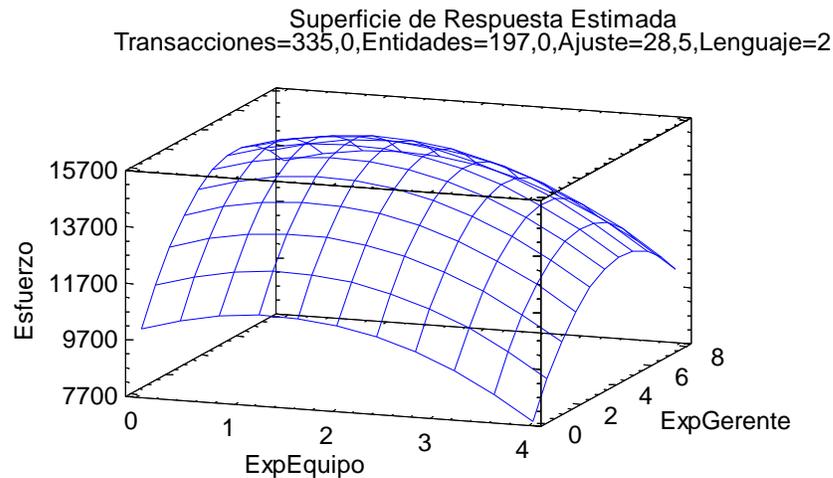
#### 4.2.Esfuerzo en función a la experiencia del gerente y del equipo

A continuación se puede observar cómo la experiencia del gerente y del equipo influencia sobre el esfuerzo necesario para el desarrollo del proyecto, según el modelo 7.

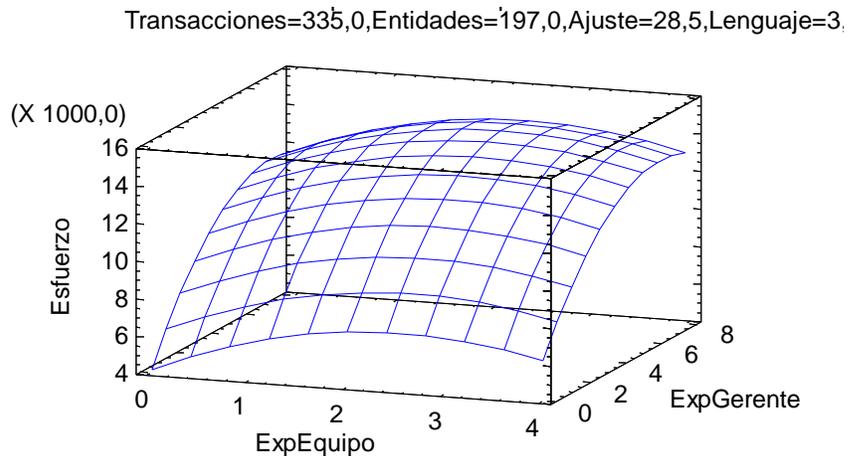


Gráfica 36. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 1, Modelo 7.

La Gráfica 36 presenta una diferencia profunda con su equivalente en el modelo número 3 (Gráfica 17), en ésta se muestra cómo la experiencia del equipo disminuye el esfuerzo, mientras que en el modelo número 3 sucede lo contrario. Por otro lado, la variable de Experiencia del Gerente tiene un comportamiento similar con el modelo 3.



Gráfica 37. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 2, Modelo 7.



Gráfica 38. Superficie de Respuesta de la Experiencia del Equipo y el Gerente sobre el Esfuerzo para el Lenguaje 3, Modelo 7.

Sin embargo, el comportamiento cambia profundamente cuando se pasa de un lenguaje a otro. Finalmente se tiene que, con un lenguaje con mayor nivel de abstracción (ver Gráfica 38), el Equipo de Proyecto requiere más tiempo. Por otro lado, se observa que el lenguaje

afecta profundamente el efecto que causa la experiencia del gerente sobre el esfuerzo, incrementando este último de una manera desmedida, en consecuencia, la experiencia del Gerente representa un factor significativo en este último modelo, incrementando el esfuerzo. Siendo el lenguaje una herramienta de programación del equipo, es extraño que éste afecte considerablemente la curva del gerente de proyecto.

El modelo número 7 tiene una arquitectura con capacidad de aprender con precisión los datos de entrenamiento, sin embargo, las inconsistencias ocurridas al cambiar el lenguaje, crea la sospecha que el modelo ha incluido en su función las dos principales debilidades de la muestra utilizada, que son:

- Falta de datos que representen por igual cada uno de los lenguajes.
- Insuficiencia de variables de entrada que justifiquen los comportamientos atípicos dentro de la muestra.

Por tal motivo, es posible que este último modelo esté incurriendo en un sobredimensionamiento y por ende un sobreajuste.

## **5. EJEMPLO PRÁCTICO DE USO**

A continuación, se presenta un caso práctico que permite ejemplificar el análisis descrito en la sección anterior. El caso consiste en el desarrollo de un proyecto de software llamado “Casa Modelo”, que se encarga de ayudar a las familias a mantener un control de los ingresos y egresos, cuentas por cobrar y cuentas por pagar dentro del núcleo familiar y a su vez permita hacer planes de inversión para futuro. Una vez realizado la definición del alcance que incluye el WBS inicial del proyecto, se procede a hacer el conteo de los elementos a desarrollar para cumplir con los requerimientos del mercado. Finalmente se concluye en el conteo de las diferentes pantallas, consultas y archivos (ver Tabla 14).

Tabla 14. Matriz para el cálculo de los puntos de función no ajustados del Proyecto “Casa Modelo”

Valor de dominio de información	Factor de ponderación						Total
	Simple		Promedio		Complejo		
	Peso	Conteo	Peso	Conteo	Peso	Conteo	
<b>Entradas externas</b>	<b>3</b>	2	<b>4</b>	7	<b>6</b>	3	54
<b>Salidas externas</b>	<b>4</b>	3	<b>5</b>	5	<b>7</b>	2	48
<b>Consultas externas</b>	<b>3</b>	6	<b>4</b>	3	<b>6</b>	1	60
<b>Archivos de lógica interna</b>	<b>7</b>	10	<b>10</b>	7	<b>15</b>	4	180
<b>Archivos de interfaz externa</b>	<b>5</b>	8	<b>7</b>	8	<b>10</b>	6	116

El siguiente paso es calcular el ajuste (indicador del grado de complejidad del proyecto) mediante la escala de likert ya predefinida en la sección 3.1.1. (ver Tabla 15).

Tabla 15. Matriz de Cálculo del Ajuste del Proyecto “Casa Modelo”

<b>Pregunta</b>	<b>Ponderación</b>
¿El sistema requiere respaldo y recuperación confiables?	2
¿Se requieren comunicaciones de datos especializados para transferir información a la aplicación, u obtenerla de ella?	0
¿Hay funciones distribuidas de procesamiento?	0
¿El desempeño es crítico?	2
¿El sistema se ejecutará en un entorno existente que tiene un uso pesado de operaciones?	0
¿El sistema requiere entrada de datos en línea?	5
¿La entrada de datos en línea requiere que la transacción de entrada se construya en varias pantallas u operaciones?	2
¿Los archivos de lógica interna se actualizan en línea?	3
¿Las entradas, las salidas, los archivos o las consultas son complejos?	2
¿Es complejo el procesamiento interno?	1
¿El código diseñado será reutilizable?	2
¿Se incluyen la conversión e instalación en el diseño?	5
¿Está diseñado el sistema para instalaciones múltiples o diferentes organizaciones?	5
¿Aplicación diseñada para cambio?	3
<b>Total de Ajuste</b>	<b>32</b>

Finalmente, el proyecto “Casa Modelo” posee los siguientes datos de entrada:

**Escenario 1:**

- Transacciones = Entradas + Salidas + Consultas = 162
- Entidades = Archivos de lógica interna + Archivos de lógica externa = 296
- Ajuste = 32
- Equipo del Proyecto = 3 años de experiencia (promedio de la experiencia del equipo)
- Gerente del Proyecto = 6 años de experiencia
- Categoría del Lenguaje = 3

Modelo 3:

Estimación del Modelo: 836 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -51,20 y 201 (ver Tabla 11) implica que el proyecto puede estar comprendido entre 734 horas/hombre y 1037 horas/hombre.

Modelo 7

Estimación del Modelo: 1143 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -134 y 118 (ver Tabla 11) implica que el proyecto puede estar comprendido entre 1009 horas/hombre y 1261 horas/hombre.

Comentario: Para este escenario, el resultado de ambos modelos son similares, incluso tienen un intervalo de solapamiento entre 1009 horas/hombres y 1037 horas/hombre. Se pudiera esperar con mayor seguridad que el esfuerzo esperado se encuentre en este último intervalo por la información que arroja los dos modelos.

### **Escenario 2:**

Partiendo del escenario 1, la junta directiva considera que es mucho tiempo para desarrollar el producto, pero se puede contar con un gerente de proyecto con mayor experiencia para lograr un desarrollo en menos tiempo. Ahora bien, cambiando el gerente de proyecto por uno de 7 años de experiencia, los resultados son los siguientes:

#### Modelo 3:

Estimación del Modelo: 772 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -51 y 201 implica que el proyecto puede estar comprendido entre 683 horas/hombre y 973 horas/hombre.

#### Modelo 7

Estimación del Modelo: 1332 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -134 y implica que el proyecto puede estar comprendido entre 1198 horas/hombre y 1450 horas/hombre.

Comentario: El modelo 3 del escenario 2 presenta una reducción con respecto al escenario 1, en este caso del 8% en el total de horas/hombres necesarias para el desarrollo del proyecto. Sin embargo el modelo 7 presenta un incremento del 16%, todo lo contrario al resultado que se pudiera esperar. Este último comportamiento ya se había observado en las Gráficas 30 y 31.

### **Escenario 3:**

Partiendo del escenario 1, la directiva de la empresa ha realizado un plan estratégico con el cual la oficina de proyectos ha decidido que “Casa Modelo” no tiene mucha prioridad con respecto a otros proyectos, por lo que se le asignará un equipo de proyectos con 2 años de experiencia y no de 3 años como estaba definido inicialmente.

#### Modelo 3:

Estimación del Modelo: 1606 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -51 y 201 y implica que el proyecto puede estar comprendido entre 1555 horas/hombre y 1807 horas/hombre.

### Modelo 7

Estimación del Modelo: 4929 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -134 y 118 implica que el proyecto puede estar comprendido entre 4795 horas/hombre y 5047 horas/hombre.

Comentario: Nuevamente el Modelo 7 genera inconsistencias al presentar un incremento exorbitante de 331% frente el modelo 7 del escenario 1. Mientras que el Modelo 3 presenta un incremento más moderado de un 40%.

### **Escenario 4:**

Partiendo del escenario 1, se ha decidido que el proyecto debe ser terminado con mayor antelación, pero no se desea cambiar al personal de proyecto. Se ha pensado en recortar el alcance del proyecto eliminando algunas funcionalidades. Finalmente se obtiene un proyecto con 146 PF en las transacciones y 266 PF en las entidades (una reducción del 10% para las transacciones y entidades respectivamente).

### Modelo 3:

Estimación del Modelo: 685 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -51 y 201, implica que el proyecto puede estar comprendido entre 634 horas/hombre y 886 horas/hombre.

### Modelo 7:

Estimación del Modelo: 916 horas/hombre

Con un intervalo de confianza de 95% y un rango de error de -134 y 118 implica que el proyecto puede estar comprendido entre 782 horas/hombre y 1034 horas/hombre.

Comentario: El modelo 3 presenta una reducción del 18% con respecto a su equivalente en el escenario 1. Mientras que el modelo 7 presenta una reducción del 20%. En este aspecto, ambos modelos se comportan en una manera consistente al compararse entre ellos.

## CONCLUSIONES

Lograr una arquitectura de RNA adecuada para resolver problemas de estimación donde se requiera la generalización constituye una tarea nada sencilla. En tal sentido, se requiere experimentar con diferentes arquitecturas y, mediante criterios bien fundamentados, seleccionar la RNA que mejor cumpla. No siempre es conveniente que la RNA aprenda con exactitud los datos de la muestra, pues a medida que aumenta su especialización, pierde su capacidad de generalizar, originándose el sobre-ajuste.

Esperar que un modelo de estimación proporcione un resultado exacto es imposible, por lo que se hace imprescindible manejar un rango de valores posibles donde se espera que se encuentre el valor real con un grado de confianza al menos de 95%. Por lo tanto, buscar que una RNA aprenda con exactitud los datos de la muestra constituye una tarea innecesaria y contraproducente.

El tamaño de la base de datos es una muestra insignificante ante la cantidad de software que es desarrollado, y a la cantidad de variables medibles que tienen una influencia significativa sobre el proyecto, por tal motivo, los resultados no son representativos ante la totalidad de la industria del software. Sin embargo, permiten establecer un procedimiento de análisis repetible en otras muestras, y así progresivamente ir comprendiendo el comportamiento de los proyectos de software.

El primer factor que influye en el incremento del esfuerzo en proyectos de software según la muestra, lo constituye el alcance del proyecto, representado por las variables Entidades, Transacciones y Ajuste en el orden que se presentan. Las Entidades siempre mantuvieron una relación lineal positiva sobre el esfuerzo, mientras que las Transacciones y el Ajuste no siempre presentaron una relación lineal. Por otro lado, la Experiencia del Equipo, luego el Lenguaje y en menor grado la Experiencia del Gerente, son variables que pueden ser

aprovechados para la disminución del esfuerzo. Estas últimas variables no afectan en la misma magnitud en que lo hace el alcance del proyecto, sin embargo, es una medida que debe ser considerada seriamente a la hora de planificar un proyecto de software.

Fue posible observar curvas cuya forma exacta son difíciles de generalizar por no contar con muestras representativas, y por depender de variables no disponibles durante la investigación, sin embargo, muchas de ellas se ajustan en gran medida a lo esperado teóricamente, lo que confirma en cierta medida la validez del modelo.

Por último, es importante concluir que, tratar de estimar el esfuerzo aplicado a proyectos, implica el estudio y utilización de varios modelos que permitan comparar resultados. Así mismo, es importante conocer en qué circunstancias el modelo presenta dificultades, y de qué manera se pueden compensar dichas debilidades, ya sea mediante otro modelo, o mediante mejoras del ya existente.

## RECOMENDACIONES

La presente investigación abre las puertas para un modelo de estudio de las curvas de aprendizaje individual y cómo afecta la productividad de los equipos de proyectos y sus gerentes. Se hace interesante determinar cómo se ve afectado el esfuerzo total de un proyecto, bajo gerentes de proyectos formados empíricamente y gerentes de proyectos formados bajo un pensum universitario. Se hace indispensable su profundización en el marco teórico, así como contar con información adicional que midan la productividad y la calidad del producto desarrollado.

Una línea de investigación más relacionada con la Ingeniería de Software, y que también es recomendable desarrollar, es la relación real existente entre las Transacciones y las Entidades. Es importante determinar cuáles son los límites coherentes en los cuales se pueden variar las Transacciones en función a un número de Entidades definidas, y viceversa. Esta delimitación permitiría realizar un análisis más focalizado en los casos que se presentan en la realidad.

Así mismo, se recomienda contar con una muestra con un número mayor de proyectos, y con datos que representen de igual forma cada uno de las categorías de lenguajes existentes.

## REFERENCIAS BIBLIOGRÁFICAS

1. AACE International. (2005). *Cost Estimate Classification System – As Applied In Engineering, Procurement, and Construction For The Process Industries*. No. 18R-97.  
[www.aacei.org/technical/rps/18r-97.pdf](http://www.aacei.org/technical/rps/18r-97.pdf)
2. Aggarwal, K.K., Singh, Y., Chandra, P., & Puri, M. (2005). *Bayesian Regularization in a Neural Network Model to Estimate Lines of Code Using Function Point*. GGS Indraprastha University, Delhi, India. Publicado en Journal of Computer Sciences.
3. Bargiela, R. A. (2007). *Monografía sobre Sistemas Expertos*. Facultad de Ciencias Exactas y Naturales y Agrimensura. Universidad Nacional del Nordeste. Corrientes Capital. Argentina.  
<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/ssee.pdf>
4. Bayona, S., Calvo, J., Cuevas, G. & San Feliu, T. (2007). *Team Software Process (TSP): Mejoras en la estimación, calidad y productividad de los equipos en la gestión del software*. Facultad de Informática Universidad Politécnica de Madrid Campus de Montegancedo, Madrid, España (Vol. 4, Nº 1, pp. 9-17).  
<http://www.aemes.org/rpm/descargar.php?volumen=4&numero=1&articulo=2>
5. Cardona, O. D. (2001). *Estimación Holística del Riesgo Sísmico utilizando sistemas dinámicos complejos*. Universidad Politécnica de Catalunya (Apéndice D. pp. 287-300).  
[http://www.tdx.cbuc.es/tesis\\_upc/available/tdx-0416102-075520//26apendiced.pdf](http://www.tdx.cbuc.es/tesis_upc/available/tdx-0416102-075520//26apendiced.pdf)
6. Carley, K., Kamneva, N. & Reminga, J. (2004). *Response Surface Methodology, CASOS Technical Report*. Carnegie Mellon University. Pittsburgh, Pensilvania.

7. Cerrillo, D. (1999). *Planificación y Gestión de Sistemas de Información. Estimación del software*. Universidad de Castilla la Mancha. Escuela Superior de Informática.  
[http://alarcos.inf-cr.uclm.es/doc/pgsi/doc/esp/T9899\\_DCerrillo.pdf](http://alarcos.inf-cr.uclm.es/doc/pgsi/doc/esp/T9899_DCerrillo.pdf)
8. Crespo, F. (2003). *Un Modelo Paramétrico Matemático Difuso para la Estimación del Esfuerzo de Desarrollo del Software*. Tesis de doctorado inédita, Departamento de Ciencias de la Computación. Escuela Politécnica, Universidad de Alcalá.  
<http://www.fdi.ucm.es/profesor/jcrespo/ante.pdf>
9. Francés, A. (2005). *Estrategia para la Empresa en América Latina*. Caracas: Ediciones IESA
10. Izaurieta, F. y Saavedra, C. (s/f). *Redes Neuronales Artificiales*. Departamento de Física, Universidad de Concepción, Concepción, Chile.  
<http://www.uta.cl/revistas/charlas/volumen16/Indice/Ch-csaavedra.pdf>
11. Lefcovich, M. (2006). *Kaizen La Mejora Continua y la Curva de Aprendizaje*.  
[http://www.wikilearning.com/curso\\_gratis/kaizen\\_y\\_la\\_curva\\_de\\_aprendizaje-kaizen\\_la\\_mejora\\_continua\\_y\\_la\\_curva\\_de\\_aprendizaje/11408-1.kaizen-curva.doc](http://www.wikilearning.com/curso_gratis/kaizen_y_la_curva_de_aprendizaje-kaizen_la_mejora_continua_y_la_curva_de_aprendizaje/11408-1.kaizen-curva.doc)
12. Leung, H., & Fan, Z. (2006). *Software Cost Estimation*. Department of Computing, The Hong Kong Polytechnic University.  
<http://www.st.cs.uni-sb.de/edu/empirical-se/2006/PDFs/leung.pdf>
13. Montaña, J. J. (2002). *Redes Neuronales Artificiales aplicadas al Análisis de Datos*. Palma de Mallorca. España.  
[http://www.tesisenxarxa.net/TESIS\\_UIB/AVAILABLE/TDX-0713104-100204/tjjmm1de1.pdf](http://www.tesisenxarxa.net/TESIS_UIB/AVAILABLE/TDX-0713104-100204/tjjmm1de1.pdf)

14. Ordieres, J.B., Torralba, J. M<sup>a</sup>. Chiner, M. (2004). *Estimación del presupuesto del proyecto de software*. Universidad de La Rioja, La Rioja, España. Universidad Politécnica de Valencia, Valencia, España.  
[http://www.aepro.com/congresos/2004\\_1/pdf/nicolas.rubio@dim.unirioja.es\\_ecdcfbee144ad7d92db4d0f0ea973b90.pdf](http://www.aepro.com/congresos/2004_1/pdf/nicolas.rubio@dim.unirioja.es_ecdcfbee144ad7d92db4d0f0ea973b90.pdf)
15. Palacios, L. E. (2005). *Gerencia de Proyectos. Un enfoque latino*. Caracas: Universidad Católica Andrés Bello.
16. Pérez, P. R. (2005). *Procesado y Optimización de Espectros Raman mediante Técnicas de Lógica Difusa: Aplicación a la identificación de Materiales Pictóricos*. Universidad Politécnica de Catalunya. (Cap. 2 Descripción general de las técnicas de la lógica difusa. pp. 35-59)  
[http://www.tesisenxarxa.net/tesis\\_upc/available/tdx-0207105-105056//04rpp04de11.pdf](http://www.tesisenxarxa.net/tesis_upc/available/tdx-0207105-105056//04rpp04de11.pdf)
17. Piorun, D. (2003). *¿Por qué fracasan los proyectos?*. degerencia.com  
Artículo consultado el día 27 de enero de 2008 de la World Wide Web:  
[http://www.degerencia.com/articulo/por\\_que\\_fracasan\\_los\\_proyectos](http://www.degerencia.com/articulo/por_que_fracasan_los_proyectos)  
<http://www.degerencia.com/dpiorun>
18. Pressman, R. S. (1998). *Ingeniería del Software. Un enfoque práctico*. (Cuarta Edición) México D.F: Editorial Mc. Graw Hill.
19. Pressman, R. S. (2001). *Ingeniería del Software. Un enfoque práctico*. (Quinta Edición) Madrid. España: Editorial Mc. Graw Hill.
20. Pressman, R. S. (2006). *Ingeniería del Software. Un enfoque práctico*. (Sexta Edición) México D.F: Editorial Mc. Graw Hill.

21. Quiros R., F. (2006). *Navegación robótica basada en aprendizaje evolutivo de acciones mediante lógica difusa*. Universidad Técnica Federico Santa María. Departamento de Electrónica. Valparaíso.  
<http://alumnos.elo.utfsm.cl/~nfeqr/descargas/Memoria.pdf>
22. Sayyad S. J. & Menzies, T.J. (2005). *The PROMISE Repository of Software Engineering Databases*. School of Information Technology and Engineering, University of Ottawa, Canada.  
<http://promise.site.uottawa.ca/SERepository>
23. Shepperd, M., & Schofield C. (1997). *Estimating Software Project Effort Using Analogies*. (Vol. 23, No. 12. pp. 736 – 743).
24. *The International Software Benchmarking Standards Group (ISBSG)*.(2008)  
Consultado el día 28 de enero de 2008 de la World Wide Web: <http://www.isbsg.org>
25. Welti S., Larissa. (2002). *Introducción al Análisis de Regresión Lineal*. Cholula, México.  
<http://web.media.mit.edu/~lwelti/documents/statistics/Analisis%20de%20Regresion.pdf>
26. Wittig, G.E. & Finnic, G.R. (1994). *Using artificial neural networks and function points to estimate 4gl software development effort*. School of Information Technology, Bond University. Gold Coast, Queensland 4229.