

**Universidad  
Tecnológica de Bolívar**  
CARTAGENA DE INDIAS

**DISEÑO DE UN MODELO PARA MEJORAR LOS PROCESOS DE ESTIMACIÓN DE COSTOS  
DE SOFTWARE PARA LAS EMPRESAS DESARROLLADORAS DE SOFTWARE**

**NOHORA NUBIA MERCADO CARUSO**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR, UTB  
FACULTAD DE INGENIERÍAS  
PROGRAMA MAESTRÍA EN INGENIERÍA. ÉNFASIS EN INGENIERÍA DE SISTEMAS Y  
COMPUTACIÓN  
CARTAGENA DE INDIAS  
2013**

**DISEÑO DE UN MODELO PARA MEJORAR LOS PROCESOS DE ESTIMACIÓN DE COSTOS  
DE SOFTWARE PARA LAS EMPRESAS DESARROLLADORAS DE SOFTWARE**

**Autor**

**Nohora N. Mercado Caruso**

Trabajo presentado para cumplir requisito al título  
Magister en Ingeniería con énfasis en Sistemas y Computación

**DIRECTOR**

**EDWIN PUERTA DEL CASTILLO**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR, UTB**

**FACULTAD DE INGENIERÍAS**

**PROGRAMA MAESTRÍA EN INGENIERÍA. ÉNFASIS EN INGENIERÍA DE SISTEMAS**

**CARTAGENA DE INDIAS**

**2013**

**Cartagena, 29 de Agosto del 2013**

**Señores:**

**COMITÉ DE EVALUACIÓN DE PROYECTOS**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

**L.C.**

Respetados Señores:

Luego de revisar el trabajo de grado "Diseño de un Modelo para mejorar los procesos de estimación de costos de software para las empresas desarrolladoras de software", de la estudiante NOHORA MERCADO CARUSO, considero que ha cumplido con los objetivos propuestos, por lo que estoy de acuerdo en presentarlo formalmente para su calificación y así optar por el título de MAGISTER EN INGENIERÍA CON ÉNFASIS EN SISTEMAS Y COMPUTACIÓN.

Atentamente,

---

**Mse. Edwin puerta del castillo**

**Cartagena, 29 Agosto del 2013**

**Señores:**

**COMITÉ DE EVALUACIÓN DE PROYECTOS  
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR  
L.C.**

Respetados Señores:

Por medio de la presente me permito presentar a ustedes, para que sea puesto a consideración, el estudio y aprobación del trabajo de grado que lleva por nombre grado "Diseño de un Modelo para mejorar los procesos de estimación de costos de software para las empresas desarrolladoras de software", del estudiante NOHORA MERCADO CARUSO, como trabajo de grado para optar el título de MAGISTER EN INGENIERÍA CON ÉNFASIS EN SISTEMAS Y COMPUTACIÓN.

Agradeciendo de antemano la atención prestada.

Atentamente,

---

**NOHORA N. MERCADO CARUSO**

**Nota de aceptación**

---

---

---

---

**Presidente del Jurado**

---

**Jurado**

---

**Jurado**

**Para mi padre, el amor de mi vida.**

## **AGRADECIMIENTOS**

Primero que todo a Dios por darme la vida.

A mis padres y mis hermanos, por ser los faros que guían quien soy.

A mi gran amigo Lucas Fontanilla por apoyarme siempre y confiar en mis capacidades.

A Ronald Zamora y Katherine Salas por luchar a mi lado durante estos dos años de estudios de la Maestría.

A mi director Edwin Puertas y demás personas que me asesoraron durante el desarrollo de esta tesis, por comprender mis inquietudes y siempre estar dispuesta a resolverlas.

## TABLA DE CONTENIDO

<b>CAPITULO I</b> .....	<b>15</b>
<b>1. DESCRIPCIÓN DEL PROYECTO</b> .....	<b>15</b>
<b>1.1. Introducción</b> .....	<b>15</b>
<b>1.2. Objetivos</b> .....	<b>16</b>
1.2.1 Objetivo General .....	16
1.2.1. Objetivos específicos .....	16
<b>1.3. Resumen de los capítulos</b> .....	<b>17</b>
<b>CAPITULO II</b> .....	<b>18</b>
<b>2. CONTEXTO Y DESCRIPCIÓN DEL PROBLEMA DE ESTUDIO</b> .....	<b>18</b>
<b>2.1. Introducción</b> .....	<b>18</b>
<b>2.2. Contexto y descripción del problema</b> .....	<b>18</b>
<b>2.3. Formulación del problema</b> .....	<b>19</b>
<b>2.4. Marco teórico</b> .....	<b>19</b>
2.4.1. Ingeniería de Software y modelos de desarrollo .....	19
2.4.2. Método de estimación juicio experto .....	21
2.4.3. Estimación por Analogía.....	23
2.4.4. Método de estimación COCOMO II.....	27
2.4.5. Método de estimación Bottom-Up.....	31
2.4.6. Precio para ganar .....	32
2.4.7. Comparación de métodos de estimación de costos de software.....	33
2.4.8. Esfuerzo y su importancia en las estimaciones de costos.....	34
2.4.9. Tamaño del Software.....	35
2.4.10. Puntos de función .....	35
<b>2.5 Estado del Arte</b> .....	<b>38</b>
2.5.1 Conceptualización de estimación de Costos de Software .....	39
2.5.2 Estudios a Nivel Mundial.....	42
2.5.3 Situación actual de Colombia .....	49
<b>2.6 Conclusión</b> .....	<b>54</b>
<b>CAPITULO III</b> .....	<b>56</b>
<b>3. METODOLOGÍA PROPUESTA PARA EL MODELO DE ESTIMACIÓN DE COSTOS DE SOFTWARE</b> .....	<b>56</b>



3.1.	Introducción .....	56
3.2	Método Delphi .....	56
3.3	El método Delphi aplicado al proyecto de Investigación .....	57
3.4	Descripción del Método Delphi .....	58
3.5	Problemas Comunes del Método Delphi .....	60
3.6	Conclusiones .....	61
<b>CAPITULO IV .....</b>		<b>62</b>
<b>4.</b>	<b>RESULTADOS DE LA INVESTIGACIÓN .....</b>	<b>62</b>
4.1	Introducción .....	62
4.2	Proceso de Investigación .....	62
4.3	Primera Ronda - La Formulación de los Escenarios .....	63
4.4	Segunda ronda: Clasificación y valoración.....	67
4.5	Tercera ronda: Confirmación de respuestas.....	67
4.6	Calificación por escenarios .....	68
4.7	Análisis de los resultados de los factores Clasificación y Valoración .....	74
4.7.1	Definición del problema y factores de clasificación.....	74
4.7.2	Definición del problema y factores de Valoración.....	80
4.7.3	Resumen de los resultados obtenidos .....	85
<b>CAPITULO V .....</b>		<b>86</b>
<b>5.</b>	<b>DISEÑO DE MODELO DE MEJORA DE PROCESOS DE ESTIMACIÓN DE COSTOS DE SOFTWARE PARA LAS EMPRESAS DESARROLLADORAS .....</b>	<b>86</b>
5.1	Introducción .....	86
5.2	Descripción de los Componentes del modelo .....	86
5.3	Ponderación de Categorías .....	92
5.4	Modelo conceptual de mejora de procesos de estimación de costos de software para las empresas desarrolladoras.....	98
<b>CAPITULO VI.....</b>		<b>101</b>
<b>1.</b>	<b>CONCLUSIONES Y TRABAJOS FUTUROS.....</b>	<b>101</b>
1.1	Conclusiones .....	101
<b>ANEXOS .....</b>		<b>108</b>
GLOSARIO DE TERMINOS .....		108
Anexo A- Perfil de los panelistas .....		109
Anexo B – Instrucciones de la encuesta ronda 1 .....		112

<b>Anexo C –Instrucciones de la encuesta ronda 2 .....</b>	<b>115</b>
<b>Anexo D –Ejemplo de envío de correo a los panelistas .....</b>	<b>120</b>

## LISTA DE FIGURAS

Figura 1. Submodelos de COCOMO II. Adaptado de (Moreno, 1995).....	27
Figura 2. Factores de Costo modelo COCOMO II. Tomado de (Barry W et al., 2000). .....	30
Figura 3. Factores de Ponderación.....	37
Figura 4. Línea de tiempo de la medición de software. Adaptado de (Mario Piattini V, 2008) .....	39
Figura 5 .Niveles de costo y personal necesario durante el ciclo de vida del proyecto. Adaptado de (Project Management Institute, 2008).....	40
Figura 6. Cono de Incertidumbre. Adaptado de (McConnell, 2006).....	41
Figura 7. Métodos de estimación usados por las organizaciones a nivel mundial .....	45
Figura 8. Resultados Obtenidos en el estudio. Adaptado de (Moløkken-Østvold et al., 2004) .....	48
Figura 9. Tipo de proyecto de Tecnología de información en Colombia. Tomado de (Alberto Cueto Vigil, 2012) .....	49
Figura 10. Distribución de sectores 2012 vs 2011. Tomado de (Alberto Cueto Vigil, 2012) .....	50
Figura 11. Presupuesto planeado de los proyectos 2010-2012 en las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012) .....	50
Figura 12. Presupuesto real 2008-2012 en las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012).....	51
Figura 13. Duración planeada de proyectos 2010-2012 en las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012) .....	52
Figura 14. Duración real 2010-2012 en los proyectos de las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012) .....	52
Figura 15. Desempeño en el cronograma en las empresas desarrolladoras de Software en Colombia. ....	53
Figura 16. Desempeño del presupuesto en las empresas desarrolladoras de Software en Colombia. Tomado de (Alberto Cueto Vigil, 2012).....	53
Figura 17. Métodos Cualitativos. ....	60
Figura 18. Grafica estadística de Medias de clasificación Vs Escenarios de Tukey HSD .....	77
Figura 19. Grafica distribución normal para residuos .....	78
Figura 20. Grafica de residuos vs valor predicho clasificación .....	79
Figura 21. Grafica de independencia clasificación.....	80
Figura 22. Grafica estadística de Medias de clasificación Vs Escenarios de Tukey HSD .....	82
Figura 23. Grafica estadística de Medias de valoración Vs Escenarios de Tukey HSD .....	82
Figura 24. Grafica distribución normal para residuos .....	83
Figura 25. Grafica de residuos vs valor predicho valoración .....	84
Figura 26. Grafica de independencia clasificación.....	85
Figura 27 Categoría del agente Metodología .....	89
Figura 28. Categoría del agente equipo de trabajo.....	90
Figura 29. Categoría del agente Herramientas Tecnológicas.....	91
Figura 30. Representación los agentes de mejora del proceso de estimación de costos.....	96

Figura 31. Categorías del modelo .....	97
--	----

## LISTA DE ECUACIONES

Ecuación 1. Distancia de Euclides.....	24
Ecuación 2. Calculo de Esfuerzo.....	28
Ecuación 3. Punto de Función.....	37

## LISTA DE TABLAS

Tabla 1. Ventajas y desventajas de los métodos de estimación. Adaptado de (Forigua y Ballesteros, 2007).....	33
Tabla 2. Cronograma de envío de Encuestas .....	62
Tabla 3: Recomendaciones y/o puntos de vista de los expertos sobre los escenarios de estimación de costos de software propuestos- Ronda 1 .....	65
Tabla 4. Escala de clasificación y valoración de cada uno de los escenarios .....	67
Tabla 5. Respuesta de experto factor de clasificación en los escenarios.....	74
Tabla 6. Resumen estadístico para la clasificación .....	76
Tabla 7. Resumen de varianza ANOVA .....	76
Tabla 8. Prueba de homogeneidad.....	77
Tabla 9. Análisis de los residuos .....	78
Tabla 10. Verificación de varianza.....	79
Tabla 11. Respuesta de valoración por los expertos en los escenarios .....	80
Tabla 12. Resumen estadístico para el factor de valoración .....	82
Tabla 13. Análisis de varianza ANOVA .....	82
Tabla 14. Análisis de Homogeneidad .....	83
Tabla 15. Análisis de residuos .....	83
Tabla 16. Prueba de varianza.....	84
Tabla 17. Variables del agente Metodología .....	92
Tabla 18. Variables del agente equipo de trabajo .....	94
Tabla 19. Variables del agente herramientas tecnológicas .....	95

## **CAPITULO I**

### **1. DESCRIPCIÓN DEL PROYECTO**

#### **1.1. Introducción**

La complejidad en los desarrollos de software denomino una “crisis del software”, que se ha tratado de resolver por medio de la introducción de métodos, metodologías, paradigmas y técnicas que tienen como fin minimizar su impacto. El objetivo no es solo que se aborde la temática de la actividad del desarrollo, sino que se trascienda a la gestión de los proyectos de software, siendo una de sus actividades la estimación de costos. (Pow-Sang Portillo, 2005)

En cualquier proyecto de desarrollo de software existen factores que afectan la producción del producto. El objetivo de la planificación del proyecto es proporcionar un marco de trabajo que permita al gestor hacer estimaciones racionales de recursos, costos y tiempo. En un proyecto de software, la estimación de costos, es una de las etapas más complejas del ciclo de vida de desarrollo. Una estimación es una predicción de cuánto tiempo durará o costará un proyecto (McConnell, 2006). Para una empresa es recomendable aplicar las diferentes técnicas de estimación en las etapas iniciales del ciclo de vida del software, sin olvidar que a medida que se avanza se va adquiriendo más conocimiento por lo que puede ser más acertada la estimación.

El autor (Capers, 2006) considera que la estimación de software es una actividad compleja, y si el proyecto es robusto se tienen que considerar muchos más factores a evaluar. La dificultad y la complejidad requerida para las estimaciones en ocasiones exceden las capacidades de los Project manager si realizan este cálculo de manera manual. Los métodos de estimación de costo de software pueden equivocarse al no tener claro los requisitos, al no existir un documento de especificación de requerimientos de sistemas, etc. Sin embargo, ningún método de estimación está completamente libre de error. En la actualidad la “mejor práctica” para la estimación de costos de software implica usar una combinación de métodos de estimación de costos de software acoplados con herramientas de gestión de proyectos.

Las empresas pueden esforzarse en desarrollar excelentes software, pero existe la necesidad de medir y controlar este esfuerzo, para no resultar en pérdida de dinero, tiempo, empleados, etc. Es por esto que este proyecto tiene como fin diseñar un modelo que permita mejorar los procesos de estimaciones de costos en las empresas desarrolladoras, teniendo en cuenta que en medida lo que garantiza que un proyecto sea exitoso es la realización de estimaciones adecuadas sobre el esfuerzo traducido en el unidad de tiempo hombre, que permitan incrementar el nivel del servicio que se presta y la calidad que se ofrece.

## **1.2. Objetivos**

### **1.2.1 Objetivo General**

- ✓ Proponer un modelo para mejorar los procesos de estimación de costos de software para las empresas desarrolladoras con el propósito de mejorar la competitividad de las mismas.

### **1.2.1. Objetivos específicos**

- ✓ Analizar los diferentes métodos de estimación de costos de software, sus ventajas y desventajas al momento de aplicarlos en un proyecto de desarrollo, así como los factores que determinan su costo para estimaciones más precisas.
- ✓ Analizar los resultados de estudios sobre cómo realizar estimaciones de costos basadas en la planificación de proyectos de software.
- ✓ Ajustar los escenarios de estimaciones de costos de software las realidades en las empresas desarrolladoras, que conlleven a una mejora en los procesos de sus estimaciones.
- ✓ Proponer un modelo de estimación de costos de software basado en los métodos más utilizados a nivel mundial adaptado a los diferentes escenarios de estimación identificados por el panel de expertos, para mejorar los procesos de estimación de costos de software en las empresas.



### 1.3. Resumen de los capítulos

El presente proyecto de investigación que tiene como objetivo mejorar los procesos de estimación de costos de software para las empresas desarrolladoras con el propósito hacerlas más competitividad y realizar estimaciones más exacta, se llevará a cabo teniendo en cuenta los siguientes aspectos metodológicos:

- ✓ Primeramente, se procederá a estudiar los diferentes métodos de estimación de costos de software, sus ventajas y desventajas. Explorar el estado de las estimaciones de software en otros países, con el fin de obtener información de posibles mejoras para los métodos y proceso de estimación de costos. Se identificará la situación actual de Colombia en cuanto a la planificación de proyectos de software, teniendo en cuenta tiempo de entrega, cronograma y presupuesto establecido.
- ✓ En el siguiente capítulo, se explicará detalladamente la metodología utilizada, que es la base para diseñar el modelo de mejora de procesos de estimación de costos de software, realizando una evaluación de los diferentes escenarios o tendencias propuestos por medio del método Delphi.
- ✓ Al recopilar la información de valoración y clasificación de todos los expertos, se realiza un análisis estadístico por medio de un diseño experimental, arrojando una tendencia y consenso de los escenarios mejor evaluados. Para comprobar el modelo estadístico, se realizaron pruebas de independencia, igualdad de varianza y análisis de residuos.
- ✓ En el quinto capítulo, se estructuran los componentes con sus respectivos factores y niveles del modelo propuesto. Se explica cómo cada uno de los componentes interactúa a lo largo de vida del desarrollo del software y se establece la matriz del modelo para identificar los niveles por factores.

## CAPITULO II

### 2. CONTEXTO Y DESCRIPCIÓN DEL PROBLEMA DE ESTUDIO

#### 2.1. Introducción

Esta sección proporciona una introducción al proyecto de investigación y a la temática de la estimación de costos de software, su importancia, ventajas de su aplicación en proyectos de desarrollos de software a nivel empresarial. Se presenta el marco teórico que soporta esta investigación y el estado del arte que evidencia los últimos avances y estudios a nivel mundial de la temática de estimación de costos de software.

#### 2.2. Contexto y descripción del problema.

La estimación de proyectos de software es una actividad compleja y crucial para las empresas de desarrollo de software. La permanencia de muchas de ellas en el mercado, depende de una buena estimación, la cual determina el éxito o fracaso de un proyecto de software al influir en todas las fases de desarrollo. El autor Heemstra (1990) señala que para muchas organizaciones es alarmante controlar el desarrollo del software; demostrando que esta es razón suficiente para que la estimación de costo ocupe un lugar importante en la disciplina de ingeniería de Software. Alguno de los factores que afectan la precisión de las estimaciones y logra que se torne un proceso difícil y complicado son:

- ✓ Datos incompletos del proyecto de software.
- ✓ Las estimaciones se hacen a toda prisa, sin tener en cuenta el esfuerzo. No hay claridad en las especificaciones de los requisitos del sistema.
- ✓ Las especificaciones claras, completas y fiables son difíciles de formular, especialmente al inicio del proyecto de software. Al ir avanzando en el desarrollo se deben ajustar los planes y presupuestos.
- ✓ Las características del software y del desarrollo de software en ocasiones son difíciles de estimar. Por ejemplo, el nivel de abstracción, complejidad, medición de productos y procesos, aspectos innovadores, etc.
- ✓ Un gran número de factores tienen influencia sobre el esfuerzo y el tiempo para desarrollar software. Ejemplo el tamaño y complejidad del software, compromiso y participación de los usuarios de la organización, la experiencia del equipo de desarrollo. En general estos factores de costos son difíciles de estimar en la operación.
- ✓ Los cambios en las tecnologías de información (TI) y las metodologías de desarrollo de software son difíciles para lograr una estimación acertada. Ejemplo, influencia de nuevos bancos de trabajo, lenguajes de cuarta y quinta generación, etc.

- ✓ El estimador calcula cuánto tiempo puede tomarse realizando una actividad, ignorando el hecho de que el subalterno cuenta con menor experiencia, generando poca productividad.
- ✓ El estimador tiende a reducir en algún grado la estimación con el fin de hacer una oferta más aceptable al cliente.
- ✓ Adicionar personal a un proyecto retrasado.

Por todo lo anterior, las empresas deben tener procesos formales y objetivos que al finalizar un proyecto permita medir los costos reales del mismo, y tener bases históricas que faciliten la utilización de esos datos para incluirlos en otros proyectos similares para sus estimaciones. Existen diferentes métodos para realizar estimaciones de costos de software y cada uno tiene su ventaja e inconveniente, por lo tanto, muchos autores recomiendan aplicar más de uno, comparar sus resultados y verificar el método más preciso.

Al hacer una revisión de la literatura no se encontró un estudio realizado en Colombia que permitiera conocer cuáles son los métodos de estimación más utilizados, cual es la opinión por los líderes de proyectos y gerentes de empresas desarrolladoras y como se aplican, esto con el fin de realizar comparaciones con investigaciones en otros países más avanzados en materia de estimación de proyectos de software.

Lo anterior son elementos de juicio que muestran la pertinencia de la realización del presente proyecto, en donde el diseño de un modelo de estimación de costos de software puede ayudar al mejoramiento de procesos de las empresas y generar una correcta planificación de proyectos de software apoyando a las empresas a identificar mejora en los procesos de estimación para incrementar los niveles de competitividad frente a empresas tanto a nivel nacional como internacional.

### **2.3. Formulación del problema**

¿Cómo el modelo de estimación de costos software puede mejorar los procesos de producción de software en las empresas dedicadas a esta actividad?

### **2.4. Marco teórico**

#### **2.4.1. Ingeniería de Software y modelos de desarrollo**

El autor Sommerville (2005) define la Ingeniería de Software como una disciplina que comprender todos los aspectos de producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después que se utiliza. En esta definición existen dos frases claves:

**Disciplina de la Ingeniería:** Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos, y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir las soluciones a los problemas, aun cuando existan teorías y métodos aplicables para resolverlos. Los ingenieros también saben que deben trabajar con restricciones financieras y organizacionales, por lo que buscan soluciones tomando en cuenta estas restricciones.

**Todos los aspectos de producción de software:** La ingeniería de software no solo comprende los procesos técnicos del desarrollo de software, sino también actividades como la gestión de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción del software.

De igual manera Pressman (2005) asegura que la Ingeniería de software es un proceso que se caracteriza por tener un marco común del proceso definiendo un pequeño número de actividades que son aplicables a todos los proyectos de software, con independencia de su tamaño o complejidad. Un número de conjuntos de tareas, donde cada uno es una colección de tareas de trabajo de ingeniería del software, hitos de proyectos, productos de trabajo, y puntos de garantía de calidad, que permiten que las actividades se adapten a las características del proyecto y a los requisitos del equipo del proyecto.

Los métodos de la ingeniería del software indican “como” construir técnicamente el software. Los métodos abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento. Los métodos de la ingeniería del software dependen de un conjunto de principios básicos que gobiernan cada área de la tecnología e incluyen actividades de modelado y otras técnicas descriptivas. Para desarrollar un producto de software se han establecido estrategias contenidas en diferentes modelos, los más utilizados en la ingeniería de software son: (Pressman, 2005).

**Modelo lineal secuencial**, llamado comúnmente ciclo de vida básico o modelo en cascada. Establece un enfoque sistemático y secuencial para el desarrollo de un software comenzando por el nivel de sistema, seguido del análisis, diseño, codificación, prueba y mantenimiento. Comprende las actividades de Ingeniería y modelado de sistemas donde se establecen los requisitos; el análisis de los requisitos que comprende la recopilación de requisitos con el fin de comprender la naturaleza del software, el diseño que tiene como función traducir los requerimientos en una representación del software. En el código se traslada todo lo diseñado al lenguaje máquina y las pruebas que se centran en los procesos internos del software, asegurando que funcionen correctamente. La fase de mantenimiento que se encarga de corregir los errores y adaptar el software a los cambios del entorno.

**Modelo de prototipo**, comienza con la recolección de requisitos, el desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un “diseño rápido”. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el cliente. Dicho diseño lleva a la construcción de un prototipo, el cual es

evaluado por el cliente y es utilizado entonces para refinar los requisitos del software a desarrollar.

Entre los modelos evolutivos se encuentra el **modelo incremental**, el cual combina elementos del modelo secuencial con un enfoque interactivo de construcción de prototipo. Este modelo aplica secuencias lineales en escalones a lo largo del proyecto generando un incremento del software. El desarrollo incremental es particularmente útil cuando la dotación de personal no está disponible para una implementación completa en la fecha límite que se haya establecido para el proyecto. Los primeros incrementos se pueden implementar con menos personas. Por otra parte el **modelo en espiral**, mezcla la naturaleza iterativa de construcción de prototipo con aspectos controlados del modelo lineal secuencial. En el modelo espiral, el software se desarrolla en una serie de versiones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones, se producen versiones cada vez más completas del sistema diseñado.

#### **2.4.2. Método de estimación juicio experto**

El autor Boehm (1984) define el juicio de experto como la consulta de uno o más expertos, donde se solicita el asesoramiento de personal capacitado. Para (Hughes, 1996) una estimación con juicio de experto se basa en la experiencia de una o más personas que están familiarizadas con el desarrollo de aplicaciones de software similares.

Este método de estimación se utiliza cuando existe ausencia de datos cuantificados. El autor Hareton Leung (2002) expresa que en este método de estimación se consultan varios expertos en técnicas de desarrollo de software y dominio de aplicaciones. Cada uno de ellos estima el costo del proyecto basado en sus métodos y experiencias. Estas estimaciones se comparan y discuten y las inconsistencias se resuelven con mecanismos como Delphi. Este método es relativamente económico y puede ser muy preciso si los expertos tienen experiencia en sistemas similares. La técnica funciona de la siguiente manera:

- ✓ El coordinador presenta a cada experto, un formulario con especificaciones para registrar las estimaciones.
- ✓ Cada experto lo diligencia en forma individual (sin discutir con los demás) y se permiten hacer las preguntas al coordinador.
- ✓ El coordinador prepara un resumen de todas las estimaciones de los expertos, se incluye la mediana de un formulario para solicitar otra iteración de las estimaciones de los expertos y la justificación para estas.
- ✓ Repita el paso 2 si es necesario.
- ✓ Tantas rondas como sea apropiado.

El juicio que emite el experto no tiene que ser informal e intuitivo. Los investigadores han encontrado diferencias significativas entre la exactitud “la opinión de expertos intuitiva” que tiende a ser inexacta y un juicio de experto estructurado, que produce estimaciones que son casi tan exactas como estimaciones basadas en modelos (Jørgensen, 2004). Es así que la fiabilidad de las estimaciones basadas en juicio de experto depende en muchas ocasiones de la experiencia y la capacidad del experto para recordar hechos de los proyectos históricos. Las estimaciones son de orden cualitativo y no objetivo. Un problema en el uso de este método es que es difícil para las personas que intervienen en el proyecto, reproducir y utilizar la experiencia de un experto (Heemstra y Kusters, 1988).

En un estudio aplicado a empresas holandesas por Heemstra y Kusters (1991) se demostró que el 61% de las organizaciones que producen desarrollo de software, basan sus estimaciones en la intuición y la experiencia y sólo el 16% utilizan modelos de estimación formales. Un estudio similar en Nueva Zelanda por Paynter (1996) reportó que el 86% de empresas de desarrollo de software que aplican métodos de estimación de expertos y solo el 26% aplican métodos automatizados.

De igual forma, investigaciones en una gran compañía financiera (Hill, et al., 2000) dieron como resultado, que los modelos más usados para realizar estimaciones son los no formales. Jørgensen (1997) al aplicar una encuesta a 26 industrias de desarrollo de software de Noruega, se reportó que el 84% de las estimaciones se basaron en juicio experto. Los autores muestran que este tipo de estimación son muy frecuentes en las empresas de software; el autor Jørgensen (2004) afirma que esta tendencia se presenta porque las organizaciones se sienten incomodas con modelos que no entienden y al aplicar este método es mucho más sencillo y flexible. A pesar de ser uno de los métodos más utilizados, muchos autores lo describen como un método intuitivo y no hay un proceso estructurado de como los expertos producen las cifras para realizar sus cálculos en las estimaciones, lo que puede generar sesgo y sensibilidad en la información. La precisión de la predicción depende de la disponibilidad de la información, y conforme como avanza el proyecto si están disponibles los datos, las estimaciones serán más fiables.

La mayoría de los pasos del método de juicio experto basan sus estimaciones en esfuerzo, como por ejemplo no culminar una actividad en un proyecto puede ser explícita y se puede revisar fácilmente. Los pasos que necesitan cuantificación se basan en alto grado en la intuición y rara vez se apoyan en la argumentación explícita y analítica. En el estudio de Jørgensen (2004) se evidencio, que al evaluar este método, se mostró que hay falta de argumentación analítica y uso frecuente de frases como “Pienso que...”, “Siento que...”; resultados similares se informaron en el estudio de estimaciones de software realizado por Mukhopadhyay, Vicinanza, y Prietula en los años de 1992: El protocolo verbal contiene poca información explícita de los procesos cognitivos implicados en la selección de un proyecto; es decir la selección de proyectos similares o análogos como base para estimaciones de nuevos proyectos.

En muchas ocasiones el módulo de un programa con alta complejidad y tamaño no se permite identificar con gran precisión la estimación de su costo, recurriéndose a la opinión de expertos, y en un análisis más sofisticado teniendo en cuenta el esfuerzo y otras variables que intervienen en el proyecto.

Los expertos al con experiencia en proyectos similares pueden obtener un grado de precisión importante en las estimaciones. Sin embargo no se recomienda solamente usar esta opción ya que se basa en opiniones humanas, por lo que se recomienda acudir a otras técnicas para completar la técnica de juicio de experto. (Zulkefli B in Mansor, 2010)

La opinión de los expertos es a menudo difícil de cuantificar, pero puede ser una herramienta efectiva de estimación en la medida al poder utilizarse para ajustar los resultados de métodos algorítmicos. La técnica de juicio experto, es mucho más sencilla para las empresas que los modelos algorítmicos, que son considerados en muchos casos complejos y difíciles de implementar. En muchas ocasiones existe escasez de datos para la calibración de los métodos de estimación, por lo que el juicio experto tiene la ventaja al introducir elementos subjetivos que permite reducir los valores atípicos que se identifiquen y que en ocasiones son difíciles de medir. (Gray, et al., 1999)

El autor Jorgensen (2005) considera que la opinión de expertos es un enfoque común, donde las empresas se ven atraídas al utilizar modelos de estimación no tan formales siendo una práctica fácil de implementar. Sin embargo es importante que las estimaciones no solo se basen en las intuiciones, siempre debe existir una justificación para que puedan ser revisadas generando mejores procesos dentro de la compañía. Una de las desventajas de esta técnica es que se depende de la experiencia del experto en recordar proyectos similares para disminuir el porcentaje de error en las estimaciones. Por lo anterior los resultados empíricos sugieren una estrecha relación entre la experiencia y la importancia de encontrar verdaderos expertos.

### **2.4.3. Estimación por Analogía**

La estimación por analogía se utiliza cuando se tiene un nuevo proyecto donde se compara con datos históricos de proyectos similares, que luego se utilizaran para predecir el costo del nuevo proyecto. Generalmente esta estimación se divide en cuatro partes: Conjunto de datos históricos, una función de correlación, una función de similitud y reglas asociadas de recuperación. Una de las partes principales es la función de similitud, que mide el nivel de exactitud entre dos proyectos diferentes. Cada parte del proyecto tiene una posición en la función de similitud y por lo tanto, determina en gran medida los proyectos históricos que se deben recuperar para la predicción final por medio de las reglas de recuperación (Kolodner, 1988).

Para el autor Li, et al. (2009) la estimación por analogía comprende cuatro etapas para su implementación:

- ✓ Recopilar información de proyectos pasados y levantamiento de datos históricos.
- ✓ Seleccionar las nuevas características relevantes del proyecto, tales como los puntos de función (PF) y líneas de código fuente (LOC) que son recogidos de los proyectos anteriores.
- ✓ Recuperación de los proyectos pasados, estimar las similitudes entre el nuevo proyecto y los proyectos anteriores y encontrar la más parecida.
- ✓ Predecir el costo del nuevo proyecto seleccionado por la función de solución. En general el promedio ponderado se utiliza como función de solución.

El conjunto de datos históricos que guarda toda la información de los proyectos anteriores es un componente clave en el sistema de estimación basado en analogía. Sin embargo, a menudo los datos pueden contener ruido o ser redundantes. Al reducir el conjunto de datos históricos en un subconjunto más pequeño pero más representativo, la selección de proyectos afecta positivamente la estimación. En primer lugar, reduce el espacio de búsqueda, por lo tanto hay más recursos informáticos para la búsqueda de proyectos similares. En segundo lugar, también produce predicciones de calidad, ya que se elimina el ruido en el conjunto de datos históricos. En la función de similitud se debe determinar el criterio para establecer las distancias entre proyectos. Existen multitud de medidas de similitud entre ejemplares, siendo las más usadas las distancias de Euclides, de Manhattan y de Minkowski.

Una de las más usadas, es la distancia de Euclides:

$$\text{Distancia } (P_x, P_y) = \sqrt{\sum_{i=1}^m (P_{xi} - P_{yi})^2}$$

**Ecuación 1. Distancia de Euclides**

La ecuación muestra el grado de similitud entre pares de proyectos usando el método de la distancia de Euclides basada en la estimación por analogía, donde m es el número de los conductores de esfuerzo,  $P_x$  es el proyecto estimado, y  $P_y$  es un proyecto basado en datos históricos.  $P_{xi}$  es el valor del esfuerzo i hallado en el proyecto x y  $P_{yi}$  es el esfuerzo i hallado en el proyecto y. La distancia entre los dos proyectos es 0 si los valores de la función de los esfuerzos calculados son los mismos. La distancia aumenta a medida que disminuye la similitud; la distancia métrica mide la similitud entre dos proyectos en términos de manejador de esfuerzo y la suma de los cuadrados del esfuerzo es la distancia total entre los proyectos. Por lo anterior, el análogo más cercano al proyecto que se estima es el proyecto con la distancia mínima euclidiana entre todos los proyectos históricos.

El número de proyectos análogos a considerar puede variar entre ciertos valores. No obstante, debe tenerse en cuenta que escoger un número muy pequeño puede conducir a considerar



solamente proyectos muy independientes del resto, y un número demasiado grande disminuye la potencia ofrecida por la analogía (Chiu y Huang, 2007). Algunos estudios afirman que no existen demasiadas diferencias en la exactitud de las estimaciones al considerar grupos de proyectos análogos de diferentes tamaños (Jeffery, et al., 2000)

El autor Li et al. (2009) ha estudiado la estimación basada en analogía e identifica la falta de exactitud del modelo y afirma que es necesario optimizar los pesos de las características del proyecto (ponderación de características) en una función similar. Propone la técnica FWPSABE (función de la ponderación y selección de los proyectos de estimación basada en la analogía) que reduce la base del proyecto entero en un subconjunto pequeño, que se aplican a proyectos representativos. Este modelo se ha validado con cuatro conjuntos de datos (dos conjuntos de datos del mundo real y dos conjunto de datos del mundo artificial) y se comparó con la estimación con analogía, demostrando que la técnica de selección de proyecto podría mejorar para la estimación de costos de software. Con este modelo se pueden producir predicciones más precisas que con otras técnicas de estimación de costo.

En 2008 se realizó un estudio combinación de regresión y estimación por analogía en modelos semi-parametricos para la estimación de costos de software. Estos dos métodos han sido analizados en cuanto a la exactitud predictiva y se han encontrado falta de convergencia entre la superioridad de un método y otro. Se combinaron los resultados de veinte estudios en los periodos comprendidos entre 1997 y 2004 y se mostraron diferencias significativas en la regresión frente a la estimación por analogía, y en otros casos similitudes entre los dos métodos. Se han discutidos están diferencias y se identifica que pueden derivarse de la falta de estandarización en la investigación de las metodologías de software que conducen a muestras heterogéneas (Mittas y Angelis, 2008).

La universidad de Singapur en el 2007, realizó un estudio de algoritmo genético para la selección de proyectos de software basado en el método por analogía en estimación de costo de software. Con este estudio se esperaba mejorar la exactitud de la predicción baja y la sensibilidad de los valores atípicos. Para mejorar estos inconvenientes se introduce la selección de subconjuntos de proyectos adecuados por el algoritmo genético. Los resultados con este método y las comparaciones con otros métodos indican eficacia y un método optimizado para la estimación de costos de software. Como trabajo el autor propone mejorar el conjunto de datos históricos y las características de los pesos ya que están optimizados por separado en este estudio, sin embargo la optimización simultánea de estos dos componentes podría conducir a una mejor predicción del rendimiento (Li, et al., 2007).

Diversos estudios demuestran que con el método de estimación por analogía los resultados son más exactos que al utilizar el método COCOMO o puntos de función, esto análisis son evidenciados por los autores (Al-Sakran, 2006), (Chiu y Huang, 2006), (Shepperd y Schofield, 1997).

El autor Álvarez Sáez (2009) resalta las ventajas de utilizar el método de estimación de costos basado en analogía, destaca que es un modelo que se basa en un razonamiento similar al de la mente humana y es apropiado para aquellas situaciones en las que el dominio es difícil de modelar. Al existir múltiples factores que influyen en la cantidad de esfuerzo para completar el proyecto, en ocasiones son se conoce como interactúan entre sí o a cual se le debe otorgar mayor peso o importancia, aquí es donde la estimación por analogía ofrece buenos resultados sin la necesidad de tener claridad de como el esfuerzo está relacionado con otros factores del proyecto.

Se puede usar teniendo un conocimiento parcial del proyecto que se quiere estimar, donde la información que más cobra importancia es la de los proyectos ya completados que se usan de consultan por medio de base de datos históricas para encontrar los más parecidos al que está siendo estimado. Por otra parte este método suaviza problemas debido a la calibración, es útil así se utilicen datos procedentes de otras organizaciones, si el proyecto análogo se encuentra entre un conjunto de datos y se midan las variables teniendo siguiendo un criterio uniforme y consistente.

Entre las desventajas se presentan que al escoger un proyecto de la base de datos histórica puede que no sea apropiado porque algunos de los factores que afectan el esfuerzo presentaron cambios radicales con el tiempo. Por lo anterior es importante el mantenimiento constante y actualización de la base de datos de proyectos.

El autor señala que al utilizarse la distancia de Euclides como criterio de selección de proyectos análogos, otorgando el mismo peso a cada factor. No existe nada que indique que este método de pesado sea el idóneo, existiendo la posibilidad de que existiendo la posibilidad de que algunos factores deban tener un mayor peso a la hora de calcular las distancias. Como alternativas la asignación de pesos encontramos el juicio experto, los coeficientes de correlación o los algoritmos genéticos.

Una vez determinado el proyecto análogo, al estimador se le plantea la cuestión de ajustar su valor de esfuerzo. De nuevo el abanico de posibles ajustes es muy amplio, por lo que la elección de uno u otro se vuelve complicada, sobre todo al tener en cuenta las diferentes características del proyecto que se quiere estimar y del análogo. Además, se pueden escoger como análogos 1, 2 e incluso 3 proyectos, ya que los resultados obtenidos en diversas experimentaciones afirman que apenas influye la elección de este número.

Por todo lo anterior, la exactitud de este método, en gran medida se apoyan en cuatro factores: la selección de un proyecto adecuado, la estrategia utilizada para encontrarlo, el tratamiento de las diferencias existentes entre el análogo y el proyecto a estimar para realizar predicciones y la fiabilidad de datos usados.(Álvarez Sáez, 2009)

#### 2.4.4. Método de estimación COCOMO II

Existen otros métodos de estimación de costo donde se utiliza la predicción o aproximación al principio del ciclo de vida del software (Mario Piattini V, 2008). El modelo COCOMO con sus siglas Constructive Cost Model (Modelo constructivo de costos) se publicó en 1981 por Barry Boehm y se basaba en las técnicas de desarrollo de esa época. Este modelo ha avanzado y se construyó uno más complejo, COCOMO II que tiene en cuenta el gasto del esfuerzo que se origina al diseñar y gestionar el proceso de desarrollo, llegando al producto de software y teniendo en cuenta practicas no tradicionales como reusabilidad, reingeniería, etc. Con el fin de apoyar los distintos sectores del mercado software, COCOMO II proporciona una familia de modelos de estimación de costo de software cada vez más detallado y tiene en cuenta las necesidades de cada sector y el tipo de información disponible para sostener la estimación del costo de software. En la figura 1 se puede observar los submodelos de submodelos del método COCOMO II, cada uno ofrece mayor fidelidad a medida que uno avanza en la planificación del proyecto y en el proceso de diseño. (Moreno, 1995).

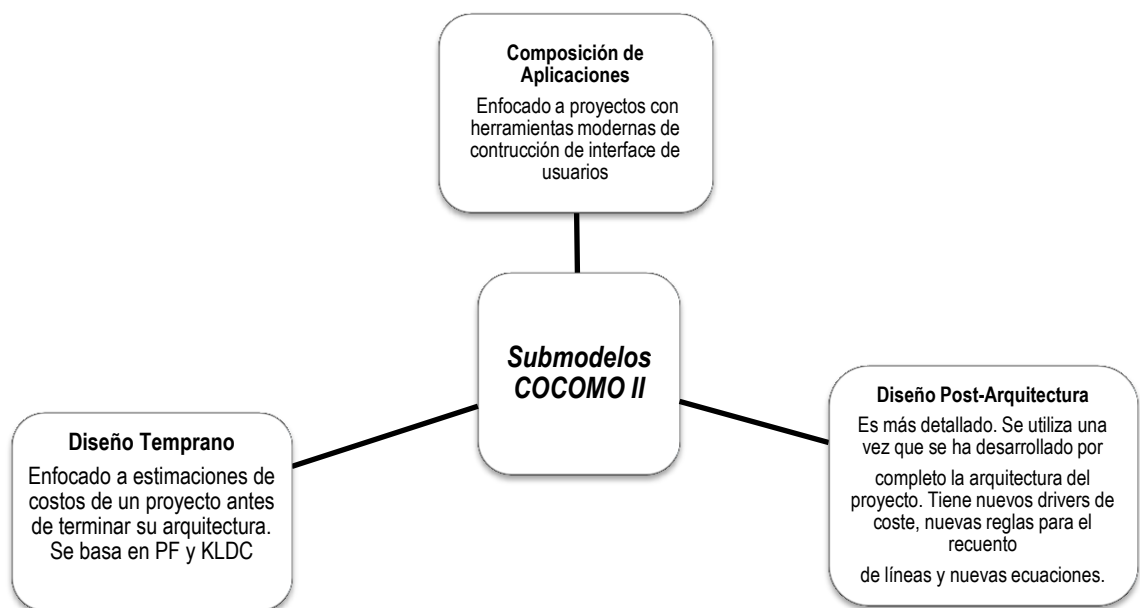


Figura 1. Submodelos de COCOMO II. Adaptado de (Moreno, 1995)

En la figura anterior se pueden resumir las características de cada submodelo, donde el modelo pos- Arquitectura cubre el desarrollo y mantenimiento, es donde se predice el esfuerzo de desarrollo, personas-mes, y utiliza 17 multiplicadores de manejadores de costo (EM). El componente de composición de aplicaciones es utilizado para estimar el esfuerzo y planificación de proyectos para un desarrollo rápido de la aplicación. El diseño temprano se usa en las etapas tempranas del proyecto y ajusta el esfuerzo nominal usando siete factores de costos.

Los siguientes son los pasos a seguir:

- ✓ **Estimación de puntos de función:** COCOMO II juega con la posible adopción de un modelo de puntos de función y uno de puntos de aplicación.
- ✓ **Conversión de puntos de función a KLDC:** En COCOMO II se abandona definitivamente la idea de medir el tamaño del código en líneas físicas y se utilizan instrucciones o líneas lógicas de código fuente. Para medir a posteriori el número de líneas lógicas.
- ✓ Aplicación de las fórmulas básicas de esfuerzo, tiempo calendario y personal requerido:

$$\text{Esfuerzo} = 2,94 \times \text{EAF} \times (\text{KLDC})^E$$

#### Ecuación 2. Calculo de Esfuerzo

En COCOMO II los factores de coste se dividen en cuatro grupos: factores de producto (Fiabilidad requerida, tamaño de la base de datos y complejidad del producto), factores de computación (restricciones de tiempo de ejecución, de almacenaje, inestabilidad de la máquina virtual y de tiempo de respuesta del equipo), factores de la plantilla (capacidad del analista, capacidad del programador, experiencia con la aplicación a desarrollar, con el lenguaje de desarrollo y la máquina virtual) y factores del proyecto (plan de desarrollo requerido, herramienta de desarrollo y la utilización de prácticas modernas de programación) (Boehm, 1981). En la década y media siguiente estas técnicas cambiaron radicalmente, por lo que la aplicación del modelo original empezó a resultar problemática y se desarrolló un nuevo modelo: COCOMO II (Sommerville, 2005).

Para los autores Boehm, et al. (1995) los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

1. Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptará tanto a las prácticas de desarrollo de la década del 90 como a las futuras.
2. Construir una base de datos de proyectos de software que permitirá la calibración continua del modelo, y así incrementar la precisión en la estimación.
3. Implementar una herramienta de software que soportará el modelo.
4. Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluarán el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

El modelo **Post- Arquitectura** es el modelo de estimación más detallado y se aplica cuando la arquitectura del proyecto está completamente definida. Este modelo se aplica durante el desarrollo y mantenimiento de productos de software en las áreas de Sistemas

Integrados, Infraestructura y generadores de aplicaciones. Se aplica en la etapa de desarrollo propiamente dicho, después que se define la arquitectura del sistema, y en la etapa de mantenimiento. Este modelo utiliza:

- ✓ Puntos de función y/o Líneas de Código Fuente para estimar tamaño, con modificadores que contemplan el rehúso, con y sin traducción automática, y el "desperdicio".
- ✓ Un conjunto de 17 atributos, denominados factores de costo, que permiten considerar características del proyecto referentes al personal, plataforma de desarrollo, etc., que tienen injerencia en los costos.
- ✓ Cinco factores que determinan un exponente, que incorpora al modelo el concepto de economía de escala. (Madachy, 1997)

Con este método de estimación se abarcan todas las tareas en relación directa a las actividades del proyecto y con todos los factores de costo de modelo, como lo es el tamaño expresado en cantidad de líneas de código fuente; el factor exponencial de escala que representa el impacto de la economía y los factores multiplicadores de esfuerzo, que simbolizan las características que influyen en el desarrollo del producto se obtiene un método de estimación completo que comienza después de la fase de revisión de requerimientos y termina con la aprobación en la fase de testeo.

Los autores Zheng, et al. (2006) estudió una extensión del método COCOMO II para el método-B, con el cual se espera obtener un modelo general en los procesos formales de desarrollo de software. El método-B se basa en la teoría, orientado al método formal para la especificación, diseño y sistemas de codificación de software. Se proporciona un marco común para la construcción de las especificaciones y mejoras en todos los niveles usando la notación de la máquina abstracta (AMN). Con este estudio se desarrolla una fórmula empírica para el modelo de post-arquitectura y da como resultado estimaciones razonables que se utilizan con el método-B.

### **El Modelo Anticipado (Pre-Arquitectura)**

El modelo de Costo de **composición de aplicaciones** utiliza los puntos objeto que son el conjunto de pantallas, informes y módulos de lenguaje de tercera generación desarrollados en la aplicación, ponderándose cada uno mediante un factor de complejidad de tres niveles (simple, medio y complejo).

La definición de los términos utilizados en los Puntos de Objetos es la siguiente:

NOP: Nuevos Puntos Objeto. (Cantidad de Puntos Objeto ajustados por la reutilización).

Srvr: Número de tablas de datos del servidor (mainframe ó equivalente) usadas junto con la

Pantalla o el informe.

Clnt: Número de tablas de datos del cliente (estación de trabajo personal) usadas junto con la pantalla o el informe.

%reuse: Porcentaje de pantallas, informes y módulos 3GL reutilizados a partir de aplicaciones anteriores prorateadas por grado de reutilización.

Tiene como entrada un conjunto de multiplicadores de esfuerzo (EM) o factores de costo que se usan para ajustar el esfuerzo nominal (PM) para refleja el producto de software que se está desarrollando. Los 17 factores generadores de costo se muestran en la siguiente figura: (Barry W, et al., 2000).

Factores	Descripción
RELY	Fiabilidad del software requerido
DATA	Tamaño de la base de datos
RUSE	Desarrollado para reutilización
DOCU	Documentación necesaria
CPLX	Complejidad del producto
TIME	Restricción de tiempo de ejecución
STOR	Restricciones de almacenamiento principal
PVOL	Volatilidad Plataforma
ACAP	Capacidad de análisis
PCAP	Capacidad de programación
APEX	Experiencia de aplicación
PLEX	Experiencia Plataforma
LTEX	Lenguaje y experiencia en la herramienta
PCON	Continuidad del personal
TOOL	Uso de Herramientas de Software
SITE	Desarrollo de Multisitios
SCED	Horario requerido de desarrollo

Figura 2. Factores de Costo modelo COCOMO II. Tomado de (Barry W et al., 2000).

Entre las ventajas que de este método, se destaca que puede ser adaptado (calibrado) a un organismo en particular, si se cuenta con la experiencia de un número importante de proyectos ya culminados que puedan aportar los datos necesarios para la calibración.

En la actualidad existen inconvenientes para realizar estimaciones precisas, pero el método COMOCO al tener trayectoria e ir evolucionando a través del tiempo ha logrado la madurez necesaria para conseguir estimaciones de gran precisión. Es importante al contar con una base de datos de proyectos que permita la calibración continua del modelo, y así incrementar la precisión de la estimación. De esta manera este modelo provee un marco analítico cuantitativo y

un conjunto de herramientas y técnicas que evalúan el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo. (Adriana Gómez y Silvina Migani, 2000)

#### **2.4.5. Método de estimación Bottom-Up**

Por otra parte con el método de Estimación Bottom-up (ascendente) el trabajo se descompone detalladamente. Se prepara un estimado de lo que se necesita para cumplir con los requisitos de cada una de las partes del trabajo, y estas estimaciones se suman luego a la cantidad total del componente del trabajo. La exactitud de la estimación ascendente se basa en el tamaño y la complejidad del trabajo identificado en los niveles inferiores (Project Management Institute, 2008). Los elementos individuales del trabajo o tareas se suman para obtener el costo total del grupo de tareas y del proyecto. La descomposición de tareas permite al administrador del proyecto asignar a los responsables de los paquetes de trabajo la estimación del costo de cada uno para después integrarlos en todo el proyecto. La estimación del costo para pequeños elementos incrementa la exactitud del estimado debido a que puede solicitarse la estimación a quién está más familiarizado. La estimación total bajo esta técnica tiende a requerir mayor tiempo para realizarse.

Según los autores Bajaj, et al. (2006) sostienen que la estimación total se basa en el conjunto de propiedades del proyecto que se descompone, generalmente por la persona responsable del desarrollo de la misma, para calcular la suma de las actividades del proyecto. Este método de estimación se utiliza para hacer una estimación global en el inicio del ciclo de vida del proyecto y es apropiado para licitaciones y contratación de actividades. Siendo fundamental para estimar el esfuerzo a nivel de trabajo y es útil para los administradores de proyecto para la planificación y gestión de proyectos en curso. Las investigaciones sugieren que un enfoque ascendente debe ser preferido en la industria a menos que la experiencia en proyectos similares sea enorme (Peters, 1999). Es muy útil cuando no existen datos históricos disponibles y cuando el proyecto es totalmente novedoso (Hughes y Cotterell, 2004).

De acuerdo con Bajaj et al. (2006), para estimar los costos de cada una de las actividades del proyecto, el método de estimación Bottom-up se basa en la estructura Work Breakdown Structure o WBS, (Estructura de descomposición de trabajo). Cada miembro es responsable de estimar el esfuerzo, la complejidad de su propio componente o modulo, y la estimación se calcula como la suma de las estimaciones de las actividades del proyecto. Con el método WBS se debe estimar las horas-hombre para cada tarea, basándose en la categoría de mano de obra; determinar el tiempo calendario requerido o disponible para cada tarea y calcular el costo real de la tarea, utilizando el costo de la categoría de mano de obra.

Los factores que impulsan el esfuerzo asociado a una tarea, se utilizan para estimar el esfuerzo general del proyecto.

- ✓ Los factores y la complejidad de los factores se estiman sobre la base de la tecnología y la experiencia previa en proyectos similares.
- ✓ Índice de productividad (horas-hombre necesarias para completar la tarea) es una estimación a partir de la experiencia de proyectos similares de los jefes de proyecto durante el desarrollo de los proyectos de software similares.
- ✓ El esfuerzo requerido para completar una tarea se calcula multiplicando una tasa de productividad específica, expresada en horas de la persona, por el número de veces que aparece el factor.
- ✓ Esfuerzo fijo.

De esta manera, la estimación de esfuerzo, ayuda a estimar el costo y el horario del proyecto. Un proyecto exitoso es el que se entrega a tiempo y dentro del presupuesto con las características deseadas. La estimación de la tasa de productividad se calcula tomando el promedio de los valores pesimista, optimista y lo más probable necesarios para completar un factor específico. Después de que una estimación se lleva a cabo, podría ser entonces necesario aplicar factor de ajuste para reflejar el efecto de factores externos no están directamente relacionado con la tarea, esto se hace de nuevo con la comparación de los proyectos similares.

Este método implicar descomponer un proyecto en partes más pequeñas para realizar estimaciones por separado, logrando que sea sencilla esta tarea y obteniendo una estimación global de todo el proyecto mucho más completa. Se basa principalmente en dos etapas, la descomposición y la integración, teniendo como ventaja que se pueden hacer explícitas muchas tareas a nivel de sistema que en ocasiones son ignoradas en otros métodos. Debido a la flexibilidad de este método los elementos se pueden estimar de forma independiente. Por ejemplo, si una parte del proyecto está estrechamente relacionado con otro proyecto recientemente desarrollado, se puede estimar por analogía, mientras que otras partes más complejas se pueden estimar por algún otro método. Sin embargo esta técnica requiere detallar los recursos utilizados y exige una especificación de requisitos, dado que estos evolucionan durante todo el proceso de desarrollo y se puede generar desperdicio de esfuerzo. De la misma manera al descomponer el proyecto y realizar las estimaciones se puede perder de vista que las partes están relacionadas como un todo, ignorando consideraciones que pueden ser evidentes al tratar al proyecto de manera global (Farrish, 2007).

#### **2.4.6. Precio para ganar**

El costo del software se estima a partir de lo que el cliente está dispuesto a pagar por el proyecto. El esfuerzo estimado depende del presupuesto del cliente y no de la funcionalidad del software. Las probabilidades de que el cliente obtenga lo que quiere es pequeña y los costos no reflejan realmente el trabajo requerido. El costo del proyecto se decide sobre la base de un esquema de propuesta y el desarrollo es limitado por el costo. (Hareton Leung, 2002)



La desventaja de este método es que se generan retrasos en la entrega al tratar de cumplir con las solicitudes del cliente, obligando al equipo de desarrollo a trabajar horas extras. Normalmente, las estimaciones de costos se basan en el presupuesto del cliente en lugar de la funcionalidad del software, donde se ajusta el precio del proyecto para mejorar la propuesta más económica realizada, con el fin de ganar el proyecto.

#### 2.4.7. Comparación de métodos de estimación de costos de software

A continuación se detallan los métodos de estimación de costos, teniendo en cuenta sus principales ventajas y limitaciones:

Método de Estimación	Ventajas	Limitaciones
COCOMO	Se basa en la evaluación de factores de esfuerzo del proyecto, lo que hace que en la estimación se incluyan varios factores que inciden en el costo del proyecto.	Predisposición por parte del equipo de la gestión ante la utilización de fórmulas matemáticas.
Estimación de expertos	Las estimaciones generadas son tan buenas como las generadas por modelos más costosos y que consumen más tiempo	Se basa en la intuición y experiencia. Se limita por la disponibilidad del experto
Analogía	Si se cuenta con buena información histórica de proyectos pasados, se pueden obtener estimaciones bastante acertadas.	Requiere contar con una base de datos histórica que sea constantemente actualizada. Compara proyectos actuales con desarrollos pasados, que en ocasiones sean desactualizados
Precio para ganar	La estimación se realiza de una manera muy sencilla.	La estimación muy probablemente estará incorrecta, y el costo real estará muy alejado de la realidad.
Bottom-Up	descomponer un proyecto en partes más pequeñas para realizar estimaciones por separado, logrando que sea sencilla esta tarea y obteniendo una estimación global de todo el proyecto mucho más completa	al descomponer el proyecto y realizar las estimaciones se puede perder de vista que las partes al estar relacionadas como un todo.

Tabla 1. Ventajas y desventajas de los métodos de estimación. Adaptado de (Forigua y Ballesteros, 2007)

#### 2.4.8. Esfuerzo y su importancia en las estimaciones de costos

El esfuerzo, según Wijayasiriwardhane, et al. (2011) es una subdisciplina de la Ingeniería de Software que ha adquirido importancia al desarrollarse software por medio de la técnica basada en Componentes (CBSD). El autor asegura que predecir con exactitud el esfuerzo al realizar un desarrollo de Software es una tarea que preocupa hoy en día a muchas organizaciones.

Es importante resaltar que las métricas recopiladas de proyectos previos sirven como base para realizar estimaciones de esfuerzo y tamaño para el trabajo de software que se esté desarrollando. Conforme el proyecto avanza las métricas se comparan con las estimaciones originales (y la planificación del proyecto). El gestor del proyecto emplea dichos datos para supervisar y controlar el progreso. (Pressman, 2005).

Por otra parte, subestimar los costos al desarrollar y realizar una mala programación puede generar un impacto perjudicial sobre la funcionalidad y la calidad del producto de software y por ende se puede ver comprometida la reputación y competitividad del desarrollador, y en situaciones extremas puede causar abandonar el proyecto. En contraste, Sobrestimar los costos y el calendario de un proyecto puede resultar una pérdida de recursos debido a la asignación redundante o perder una licitación o un contrato del desarrollo de un proyecto (Shepperd, May 2007). De esta manera, el 53% de los productos de Software no se entregan a tiempo ni dentro del presupuesto inicial, mientras el 18% de los proyectos son abandonados por completo, por lo que es importante enfatizar en la importancia del esfuerzo para predecir la exactitud del desarrollo de software. Cabe anotar que en los últimos 20 años son pocas las mejoras en la precisiones de estimaciones de costos de software, y se cree que una de las razones es la imprecisión en el uso de la terminología para la estimación de esfuerzo (Group, 2004) y (Grimstad, et al., 2006).

Dentro de los diferentes modelos de estimación de costos, el tiempo y el esfuerzo se representan por medio de una función con un número de variables. La entrada son los costos que se manejan en el proyecto, los cuales varían de acuerdo al modelo empleado. La salida es la estimación de esfuerzo y la duración en tiempo; para cada modelo se utilizan diferentes tipos de parámetro y algoritmos y se sugiere que las empresas que apliquen por primera vez un método de estimación de costos, deben realizar calibración. Para hacer posible la calibración los datos de proyectos históricos deben estar disponibles en la organización, por lo cual es importante que las empresas manejen una base de datos de proyectos pasados. De esta manera es importante tener en cuenta los siguientes pasos para aplicar un modelo de estimación de costos: (Heemstra, 1990)

- ✓ Calibrar el modelo.
- ✓ Estimar el tamaño del software (líneas de código, puntos de función, etc)

- ✓ Traducir la estimación del tamaño del software en una estimación de esfuerzo. Normalmente se utiliza la productividad para realizar esta traducción. El esfuerzo= $a \cdot (\text{tamaño})^b$ , donde las variables a y b se obtienen de la media de la regresión del análisis de proyectos anteriores.
- ✓ Ajuste del esfuerzo estimado teniendo en cuenta los factores de costo. Esfuerzo= $a \cdot ((\text{tamaño})^b) \cdot \text{factor de corrección}$ . El factor de corrección es la suma de la influencia de todos los factores de costo en el esfuerzo del desarrollo.
- ✓ Dividir el esfuerzo total sobre las fases y actividades del proyecto. Thibodeau and Dodson en 1981 propusieron dividir el esfuerzo en un gran número de proyectos. La conclusión fue que en mucho de los proyectos la división de esfuerzo se da en las fases de análisis / diseño en un 40%, codificación 20% y pruebas 40%.
- ✓ Estimar la duración. Existe una estrecha relación entre duración y esfuerzo. Duración= $a \cdot (\text{Esfuerzo})^b$ ; donde los valores de a y b se obtiene de forma empírica de proyectos anteriores. De acuerdo a Birrell and Ould (1985) el valor de a esta entre 2 y 4 y de b entre 0.25 y 0.4.
- ✓ Ejecutar el análisis de sensibilidad y riesgo, especialmente al inicio del proyecto cuando el grado de incertidumbre es grande.

#### **2.4.9. Tamaño del Software**

En un proyecto de software el tamaño es indispensable para realizar estimaciones de costo, y debe estar relacionado con el esfuerzo de las personas que realizan el desarrollo, considerándose en las métricas de tamaño los aspectos que influyen en las estimaciones de costos, como lo son la tecnología ( los métodos y herramientas de Ingeniería de Software), la complejidad ( puede tener un impacto sustancial sobre la calidad y el desempeño de equipo) y los recursos con los que cuenta la compañía.

El objetivo de las métricas es minimizar el tiempo de desarrollo haciendo ajustes para evitar demoras y reducir los problemas y riesgos potenciales; también se utiliza para valorar la calidad del producto, conduciendo a la reducción de costos globales del proyecto. (Pressman, 2005). Ya sea que se decida comprar un paquete o que se haga un desarrollo a la medida, interno o subcontratado, las organizaciones necesitan controlar esta inversión. En ese sentido, el medir el software no es un tema académico, sino un tema de valor, de inversión, de negocio (Durán Rubio, 2003).

#### **2.4.10. Puntos de función**

Según el autor Durán Rubio (2003), son muchas las empresas que cuentan con una vaga idea de la funcionalidad del software con la que cuentan y mucho menos cuánto de esa funcionalidad realmente se aprovecha. Tampoco cuentan con la productividad de sus programadores en términos de cuánta funcionalidad pueden desarrollar por unidad de tiempo. Esto les impide

estimar, con fundamento, la inversión y el tiempo necesario para que su personal pueda desarrollar un proyecto de Software o para evaluar la propuesta de un externo. En consecuencia, los retrasos en los proyectos así como los excesos en presupuesto son normales.

Las métricas del software orientadas a la función utilizan una medida de la funcionalidad entregada por la aplicación como un valor de normalización. Ya que la “funcionalidad” no se puede medir directamente, se debe derivar indirectamente mediante otras medidas directas. Las métricas orientadas a la función fueron propuestas por primera vez por Albretch, quien sugirió una medida llamada punto de función. Los puntos de función se derivan con una relación empírica según las medidas contables (directas) del dominio de información del software y las evaluaciones de la complejidad del software (Pressman, 2002).

El análisis por puntos de función es un método para cuantificar el tamaño y la complejidad de un sistema software en términos de las funciones de usuario que este desarrolla (o desarrollará). Esto hace que la medida sea independiente del lenguaje o herramienta utilizada en el desarrollo del proyecto (Albrecht y Gaffney, 1983). Según (Symons, 1988) este método sirve para medir específicamente aplicaciones de negocios y no se recomienda su uso en aplicaciones técnicas o de carácter científico.

Los puntos de función se calculan completando la tabla de la figura 3. Se determinan cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla. Los valores de los dominios de información se definen de la forma siguiente: (Pressman, 2005)

- ✓ **Número de entradas externas (EE).** Cada entrada externa se origina en un usuario o es transmitida desde otra aplicación y proporciona distintos orientados a la aplicación o información de control. Las entradas suelen emplearse para actualizar archivos lógicos internos (ALI). Las entradas deben distinguirse de las consultas, que se cuentan por separado.
- ✓ **Número de salidas externas (SE).** Cada salida externa se deriva en el interior de la aplicación y proporciona información al usuario. En este contexto, la salida externa alude a informes, pantallas, mensaje de error, etc. Los elementos de daños individuales dentro de un informe no se cuentan por separado.
- ✓ **Número de consultas externas (CE).** Una consulta externa se define como la entrada en línea que lleva a la generación de alguna respuesta inmediata por parte del software, en la forma de salida en línea (a menudo recuperada de un ALI).
- ✓ **Número de archivos lógicos internos (ALI).** Cada archivo lógico interno es un agrupamiento lógico de datos que reside dentro de los límites de las aplicaciones que se mantienen mediante entradas externas.
- ✓ **Numero de archivos de interfaz externos (AIE).** Cada archivo de interfaz externo es un agrupamiento lógico de datos externos a la aplicación, pero que proporciona datos que podrían usarse en ésta.

Valor del dominio de información	Conteo	Factor de ponderación			
		Simple	Promedio	Complejo	
Entradas externas (EE)	<input type="text"/> ×	3	4	6	= <input type="text"/>
Salidas externas (SE)	<input type="text"/> ×	4	5	7	= <input type="text"/>
Consultas externas (CE)	<input type="text"/> ×	3	4	6	= <input type="text"/>
Archivos de lógica interna (ALI)	<input type="text"/> ×	7	10	15	= <input type="text"/>
Archivos de interfaz externa (AIE)	<input type="text"/> ×	5	7	10	= <input type="text"/>
Total de conteos	→				<input type="text"/>

Figura 3. Factores de Ponderación

Al completar los datos se asocia un valor de complejidad por cada conteo. Las empresas desarrollan criterios para determinar si una entrada es simple, promedio o compleja, por lo cual la complejidad es subjetiva. De esta manera para calcular los puntos de función se utiliza la siguiente relación:

$$PF = \text{Conteo Total} \times [0,65 + 0,01 \times \sum(F_i)]$$

**Ecuación 3. Punto de Función**

Donde el conteo total es la suma de todas las entradas de PF obtenidas en la figura No 1,  $F_i$  (i=1 a 14), que son factores de ajuste de valor basados en las respuestas a las siguientes preguntas:

1. ¿El sistema requiere respaldo y recuperación confiable?
2. ¿Se requieren comunicaciones de datos especializadas para transferir información a la aplicación, u obtenerla de ellas?
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 en aplicaciones?
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 ALI 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 incluye la conversión e instalación en el diseño?
13. ¿Está diseñado el sistema instalaciones múltiples en diferentes organizaciones?
14. ¿La aplicación está diseñada para facilitar el cambio y para que el usuario lo use fácilmente?

Estas preguntas se responden empleando una escala de 0 (no importante o aplicable) a 5 (absolutamente esencial). Los valores constantes de la ecuación 3 y los factores de peso que se aplican en los conteos del dominio de la información se determinan empíricamente.

El número obtenido en PF representa la cantidad de unidades de software de un proyecto o aplicación. Con este valor podremos realizar estimaciones de plazos, costos, recursos, así como también: (Durán Rubio, 2003)

- ✓ **Administrar la productividad.** Si se tiene el tamaño y por otro lado el esfuerzo requeridos entonces se pueden establecer indicadores de productividad en términos de Horas por Puntos de Función. Este indicador puede servir para controlar un plan de mejora y para realizar comparaciones con la industria.
- ✓ **Administración de calidad.** De manera similar, si se tiene el tamaño y el número de defectos que se entregaron en un desarrollo, se puede establecer un indicador de calidad como defectos por Puntos Función, para la administración de la calidad. Con una historia suficiente de proyectos, se pueden estimar proyectos futuros.
- ✓ **Comparabilidad.** Al utilizarse la misma métrica en todos los proyectos, entonces se pueden comparar entre ellos, pero mejor aún, si es una métrica estándar en la industria podría compararse contra otros.

## 2.5 Estado del Arte

La calidad es un factor diferenciador entre las empresas que compiten por lograr un posicionamiento en el mercado. La calidad para el autor Galbraith (1974) es concebida como una relación entre la funcionalidad (lo que debe hacer el sistema) y el rendimiento (que tan bien el sistema debe hacerlo). Cuando un proyecto amenaza con exceder el presupuesto estipulado y el tiempo de entrega, la percepción de calidad para el cliente se deteriora.

Es así que la industria del Software, a pesar de su importancia en la sistematización de procesos, sigue buscando la solución para desarrollar más rápido y con mejor calidad. Un factor que dificulta esta labor es la estimación de costos y los plazos realistas. De esta manera la estimación de costo de software y, por lo tanto, la exactitud de esta, se ha convertido en un aspecto crucial tanto para la compañía desarrolladora del proyecto como para el cliente, el cual espera que el costo del software coincida lo más posible con el valor esperado para este. Una estimación más cercana a la realidad permite a la compañía una planificación en la cual se puedan utilizar mejor los recursos, siendo cada día más relevante dentro de la ingeniería de software (Joseba Esteban López, 2008). La necesidad de estimar, se evidencia en un informe estadístico en Canadá (Chaos, 2001) de 30,000 aplicaciones de proyectos de desarrollo, el 23% presentó fallas, el 49% estaba siendo cuestionado y solo el 28% fue exitoso. Estos datos son alarmantes ya que ocasionan que las empresas fracasen financieramente y se afecte la reputación de las mismas.

Existen diferentes métodos para estimar las variables para la planificación, seguimiento y control de los proyectos de software como son tamaño, esfuerzo, duración y costo. El objetivo de una estimación no es predecir el resultado del proyecto, es determinar si los objetivos son suficientemente realistas para permitir que el proyecto sea controlado para alcanzar dichos objetivos. De esta manera estimar ayuda a las organizaciones a tomar mejores decisiones del negocio, conociendo la variabilidad del proyecto de acuerdo a restricciones de tiempo, de costo y funcionalidad y saber con qué recursos se cuenta. Con las estimaciones de software se guardan datos históricos y se evalúan futuros desarrollos, si se realizan consecutivamente ayudan a conocer el avance del proyecto permitiendo ajustar el calendario y los recursos financieros de la empresa (Mario Piattini V, 2008).

### 2.5.1 Conceptualización de estimación de Costos de Software

A partir de las consideraciones del autor (Mario Piattini V, 2008), la medición de software se sitúa a finales de los 60, donde se establecen las bases para los trabajos actuales en esta área específica. En la figura 4. Se puede observar una línea de tiempo donde se realiza un recorrido a través de la medición de software desde los años 60, hasta el siglo XX:

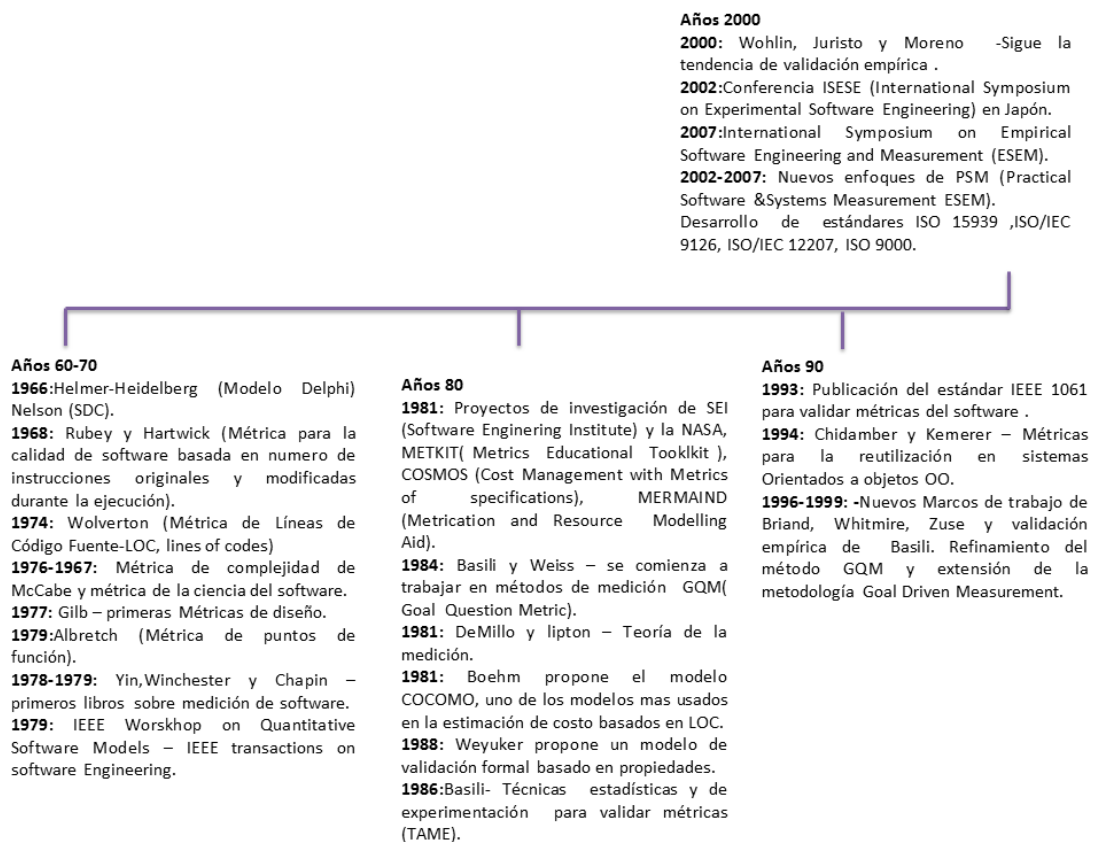
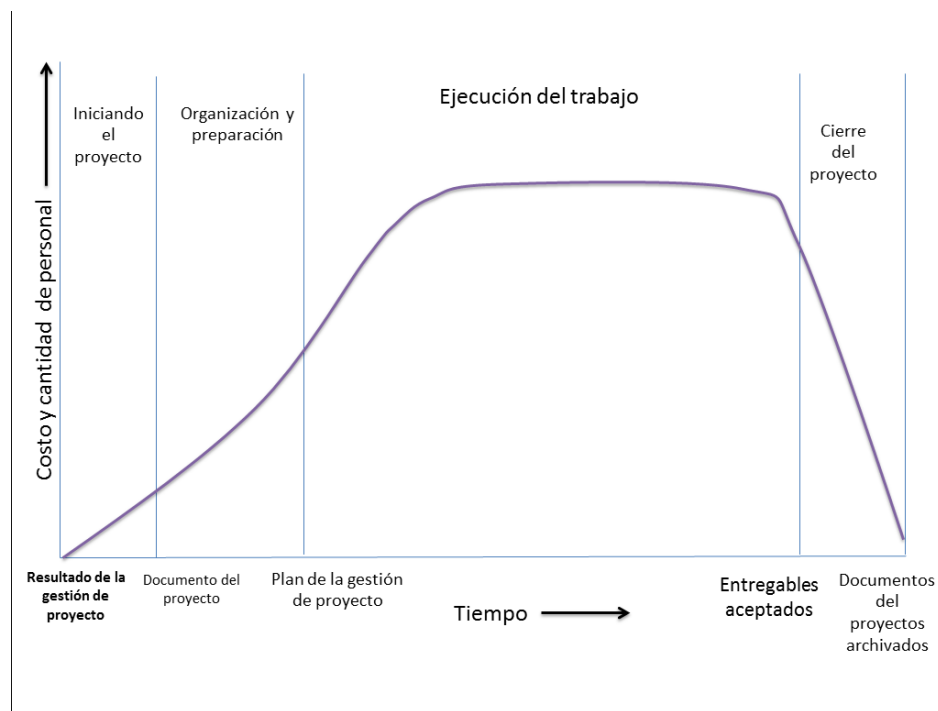


Figura 4. Línea de tiempo de la medición de software. Adaptado de (Mario Piattini V, 2008)

Según Project Management Institute (2008), un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos indica un principio y un final definidos. El final se alcanza cuando se logran los objetivos del proyecto o cuando se termina el proyecto porque sus objetivos no pudieron ser alcanzados, o simplemente cuando ya no existe la necesidad que dio origen al proyecto. Cualquier proyecto consta de un ciclo de vida que contiene un conjunto de fases que generalmente son secuenciales y en ocasiones superpuestas, cuyo nombre y número se determinan por las necesidades de gestión y control de la organización u organizaciones que participan en el proyecto, la naturaleza propia del proyecto y su área de aplicación. El ciclo de vida del proyecto puede ser determinado o conformado por los aspectos únicos de la organización, de la industria o de la tecnología empleada. Mientras que cada proyecto tiene un inicio y un final definidos, los entregables específicos y las actividades que se llevan a cabo entre éstos variarán ampliamente de acuerdo con el proyecto. El ciclo de vida proporciona el marco de referencia básico para dirigir el proyecto, independientemente del trabajo específico involucrado. De esta manera los proyectos varían en tamaño y complejidad y se pueden estructurar dentro de la siguiente gráfica:



**Figura 5 .Niveles de costo y personal necesario durante el ciclo de vida del proyecto. Adaptado de (Project Management Institute, 2008).**

En la gráfica 5 se da cuenta que la cantidad de personal un proyecto aumenta a medida que las fases del ciclo de vida se completan, lo cual incide directamente en el aumento de costos del mismo. Dicho incremento en los costos debe ser estimado y regulado a partir de la utilización de herramientas y métodos.



La necesidad de la precisión al estimar es abordado por Laird Linda M (2006), que asegura que “debe ir avanzando a lo largo del proyecto, donde es necesario dentro de un desarrollo tradicional realizar estimaciones en tres momentos: En la etapa de estudio de viabilidad o inicio del proyecto (Macro estimación), en la etapa de requisitos (Estimación detallada) y en la etapa de diseño (Estimación refinada)“ Si se estima durante la ejecución del proyecto se tienen medidas actualizadas sobre la capacidad del proceso en vez de utilizar medidas estimadas obtenidas al principio del proyecto. También las restricciones pueden haber cambiado lo que daría la oportunidad de priorizar mejoras y funciones en el sistema o aplicación que se esté construyendo.

Al momento de estimar se está proyectando una acción, pero siempre existe un grado de incertidumbre que hay que considerar. En un proyecto de software se tienen que tomar múltiples decisiones, que con las estimaciones son más confiables y por ende se reduce el nivel de incertidumbre.

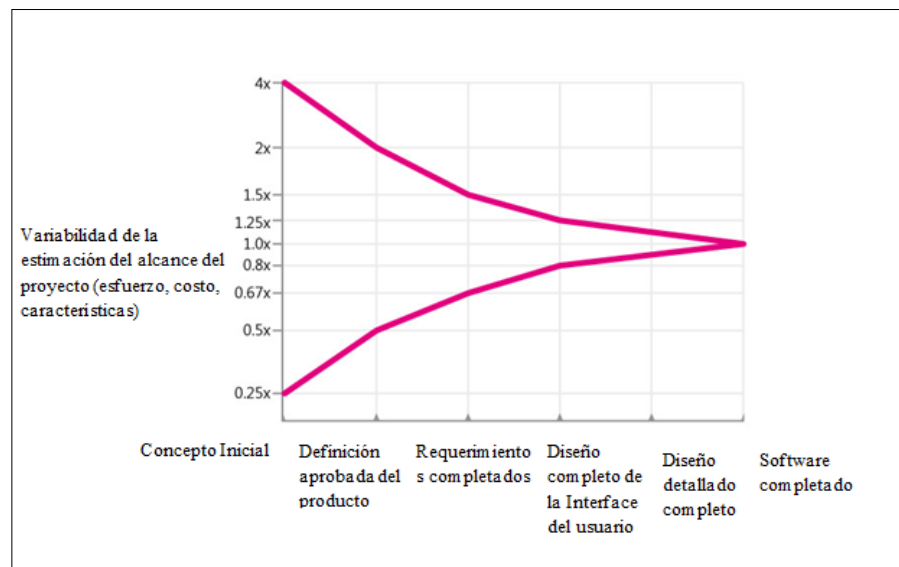


Figura 6. Cono de Incertidumbre. Adaptado de (McConnell, 2006)

En la figura 6 se puede apreciar que las estimaciones son más precisas mientras se avanza en el desarrollo del proyecto. Esta figura, propuesta por Boehm et al. (1995), muestra en el eje horizontal la variabilidad de la estimación y en el eje vertical como se disminuye el cono de incertidumbre (grado de error en las estimaciones) al hacerse más exacta la estimación del proyecto. En la gráfica “x”, representa las variables que se estiman (esfuerzo, duración, costo, etcétera).

El concepto de estimación de costo de software, el autor Felix (1997) lo define como el proceso de predecir el esfuerzo requerido para el desarrollo de un sistema de software, donde la mayor parte del costo de desarrollo es debido al esfuerzo humano, indica una complejidad en el desarrollo del proyecto de software. En ocasiones no se cuenta con una base de datos histórica,

no hay entrenamiento del personal para realizar las estimaciones correspondientes y los factores implicados no están bien interrelacionados. La precisión de la estimación del costo de software tiene un impacto directo y significativo sobre la calidad de las decisiones de inversión del software, significando pérdida o ganancia para la empresa desarrolladora (Al-Sakran, 2006).

La estimación de costos, está asociada con la estimación fiable del tamaño y los recursos necesarios para producir el producto de software que proporciona al administrador de proyecto la información necesaria para desarrollar la programación, presupuesto y asignación de personal y recursos. De esta manera el objetivo de la estimación de costo de software consiste en estimar el tamaño, el esfuerzo, la complejidad y el costo del proyecto de software para poder encontrar la mejor decisión de desarrollo y asegurar que el gasto se encuentre de acuerdo a lo presupuestado. (Bozhikova y Stoeva, 2010). Por todo lo anterior, la estimación de costos de software es importante para la planificación, programación, presupuesto y establecer el precio indicado al desarrollo del software (Magazinius y Feldt, 2010), otros autores como Pendharkar, et al. (2003) y Martin Shepperd (2007) convergen con estos principios al asegurar que es fundamental para el éxito de la gestión del proyecto de software, al afectar la mayoría de las actividades de gestión incluyendo la asignación de recursos, la licitación de proyectos y la planificación.

Al iniciar un proyecto es complejo generar una estimación de costo precisa al incluir nuevos programadores, nuevas tecnologías y equipos para desarrollar un sistema de software, pero a lo largo del proyecto las estimaciones van mejorando. No se puede afirmar cual método de estimación es mejor, por esto es conveniente combinar varias técnicas y comparar los resultados.

Los autores Genshowsky, et al. (2005) confirman que para realizar estimaciones seguras de costos y esfuerzos se puede recurrir a varias opciones posibles: Utilizar “técnicas de descomposición” relativamente sencillas para generar las estimaciones de costo y de esfuerzo del proyecto, desarrollar un modelo empírico para el cálculo de costos y esfuerzos, adquirir una o varias herramientas automáticas de estimación.

### **2.5.2 Estudios a Nivel Mundial**

La estimación de costos de Software es una herramienta útil en las empresas para lograr tiempos más reales en los desarrollos. Para obtener estimaciones más exactas es importante investigar las áreas de mejoras identificando los métodos más utilizados, la precisión de cada uno, y los factores que influyen en escoger determinado método en las industrias de desarrollo de software.

En 2008 en el estudio titulado “ Una encuesta sobre la estimación de costos de software en la industria de software chino” (Da Yang, et al., 2008) se evaluaron los diferentes métodos de estimación de costos mediante el análisis de 112 proyectos de software de un conjunto de

bases de datos y se aplicó un modelo de encuesta a 116 organizaciones, dando como resultado sugerencias basadas en evidencias de cómo mejorar las estimaciones en proyectos de software. Estudios similares se realizaron en los años 80, pero el entorno empresarial, las tecnologías y los procesos de desarrollo han evolucionado, lo que motivó a llevar a realizar a los autores esta investigación. Con la encuesta aplicada se analizó la magnitud de la distribución de la precisión de la estimación de costos en la industria china, las posibles causas de estimaciones inexactas y se exploró teniendo en cuenta datos históricos, porque muchas organizaciones no utilizan modelos de estimación formales como COCOMO, métricas FPA o algún tipo de algoritmo impulsado en estos métodos. Esta investigación tomó como referente estudios pasados para determinar tendencias y analizar la información obtenida. Se compararon los resultados con investigaciones entre 1984 y 2004 desarrolladas en diferentes países del mundo como Noruega, Nueva Zelanda, Canadá, etc.

Este estudio se basó en responder las siguientes preguntas relacionadas con la estimación de costos de software:

- RQ1: ¿Cuál es la exactitud de la estimación de esfuerzo y duración?
- RQ2: ¿El tamaño del proyecto afectará la estimación del esfuerzo y de calendario?
- PI3: ¿En qué medida son usados los diferentes métodos de estimación en la industria?
- RQ4: ¿Para qué fines se utilizan las estimaciones de costos?
- RQ5: ¿Para las personas que tan importante es la estimación, en comparación con otros aspectos del desarrollo?
- RQ6: ¿Cuán satisfecho están las personal con el costo actual del software estimado?
- RQ7: ¿Cuándo las organizaciones suelen hacer estimaciones de costos?
- RQ8: ¿Cuáles son las causas de las estimaciones inexactas?
- RQ9: ¿Cuáles son las barreras y dificultades en la aplicación de software de los modelos de estimación de costos?

Con el fin de generar un estudio completo y estructurado Kasunic (2005) sugiere 7 pasos señalados en el SEI (orientación de diseño de encuestas):

- ✓ Identificar los objetivos de la investigación. Explorar el estado de la estimación de costos de software obteniendo información de las posibles mejorar de los métodos y procesos de estimación.
- ✓ Identificar y caracterizar el público objetivo.
- ✓ Diseñar un plan de muestreo. Se utilizó a las organizaciones participantes en la “Conferencia de mejoras de procesos de software chino 2007”, la mayoría son miembros de la CSBSG.
- ✓ Diseñar y escribir el cuestionario.
- ✓ Prueba preliminar del cuestionario. Se realizó un estudio piloto en un desarrollo CMMI-4 validando el cuestionario.
- ✓ Discutir el cuestionario. Se distribuyeron 4000 copias en la conferencia de mejoras de procesos.

- ✓ Analizar los resultados y escribir un informe. El resultado final fueron 171 encuestados de 116 organizaciones, que corresponde a empresas medianas y la mayoría se encuentra en procesos de mejoras en CMMI o ISO-9000.

Para responder las preguntas, la información se clasificó en dos tipos: objetiva por los datos extraídos de proyectos históricos, y subjetiva por medio de entrevista presenciales con los encuestados. El estándar chino de software Benchmarking Group (CSBSG), proporcionó un conjunto de proyectos confiables para investigar RQ1 y RQ2. Su objetivo fue construir un gran proyecto con datos de software que reflejaran las mejores prácticas de estimación en la industria del software chino, basándose en 1012 datos de proyectos a partir de 141 organizaciones. En el estudio se utilizaron 112 proyectos de 61 organizaciones que abarcan zonas de negocio como finanzas, manufactura, gobierno, comunicación, energía, salud, etcétera.

Con esta investigación se encontró que la estimación de costos se utiliza fundamentalmente para proyectos a nivel de planificación y control, con un porcentaje del 84% que corresponden a 97 organizaciones entrevistadas, seguida de otros propósitos de uso como lo son realización de presupuestos 74%, evaluación de propuesta de proyectos, negociación de contratos, planificación a corto y mediano plazo y para la mejora de procesos de software. Lo anterior indica que los métodos de estimación se utilizan para diversos propósitos, estos pueden ser mejorados para aportar más valor a las organizaciones.

La estimación de costos en China es considerada como “importante” o “muy importante” con un porcentaje del 84%, que es similar en estudios anteriores con autores como (Lederer y Prasad, 1992) y (Moløkken-Østfold, et al., 2004). Un aspecto crucial en este estudio fue identificar la satisfacción de la estimación de costos en las organizaciones. Se identificó que el 62% se muestra neutral, 16% “satisfactorio” y 22% seleccionó “muy insatisfactorio” o “insatisfactorio”. Igualmente el estudio arrojó que las organizaciones usualmente realizan estimaciones en la fase inicial de la propuesta (57%), en el estudio de factibilidad (67%) y en la fase de requerimientos (74%), lo cual va en concordancia con la literatura, en las estimaciones hay muchos riesgos e incertidumbres al inicio del proyecto de software y al ir avanzando en el proyecto serán más exactas, reduciendo el cono de incertidumbre. Entre las causas identificadas por las empresas chinas para generar una mala estimación se pueden resaltar los requisitos volátiles y poco claros (58%), la falta de una inversión por los directivos para lograr mejorar la estimación de costos (44%) y 61 personas identificaron que los modelos de estimación no generan suficientes beneficios (40%).

En este estudio se confirmó que las empresas utilizan las combinaciones de uno o más métodos a lo largo del desarrollo del proyecto de software. Los métodos de estimación reportados son juicio de experto, precio para ganar, estimación por analogía y métodos basados en modelos. Los más usados son juicio de experto y estimación por analogía con un porcentaje del 70% cada uno, seguido de precio para ganar (53%) y solo un 15% reporto el uso de estimaciones basadas en métodos como COCOMO II. El porcentaje de la utilización de estos métodos se comparó con estudios anteriores:

Año	(McAulay, 1987)	1989(Heemstra, 1990)	(Wydenbach, 1995)	(Moløkken-Østvold et al., 2004)	2007 (Nuevo)
Porcentaje utilizado para cada uno de los métodos(más de un posible método)					
Consultor experto		26%	86%	100%	70%
Intuición y experiencia	85%	62%			
Analogía		61%	65%	28%	70%
Modelo de costos de Software	13%	14%	26%		15%
Precio para ganar		8%	16%		53%
Capacidad Relativa		21%	11%		
Top Down			13%		
Bottom-up			51%		
Otros	12%	9%	0%		

Figura 7. Métodos de estimación usados por las organizaciones a nivel mundial

Solo el 15% utilizó modelos de costos de software que es similar a los porcentajes de estudios anteriores. En todos los estudios se muestra que los métodos más usados son juicio experto y estimación por analogía. Se puede notar que 53% manejo precio para ganar y capacidad relativa, que en el 2004 ya no se utilizaba y en años anteriores el porcentaje de uso fue muy bajo. Boehm argumenta que los métodos relacionados con la capacidad relativa las malas prácticas y el precio para ganar generalmente producen excesos en las estimaciones, por lo que estos métodos son inaceptables (Boehm, 1981) ya que depende del presupuesto del cliente y no de la funcionalidad del software. Como trabajo futuro se recomienda mejorar los proceso de desarrollo de software para disminuir la incertidumbre en la estimación de costos, y realizar seguimiento al desarrollo de software en China a través del tiempo para analizar la varianza y los cambios en el desempeño de las estimaciones y las posibles mejoras a estos.

En conclusión, en esta investigación se evidencio que los proyectos robustos tienen más exceso de esfuerzo que los pequeños y que tienen mayor varianza de precisión presentando problemas de mayor esfuerzo en la etapa de programación. Es importante identificar cuáles son los fines de realizar estimaciones de costos en los proyectos de software. De 115 organizaciones entrevistadas, se resalta el uso de técnicas de estimación en la etapa de planificación y control del proyecto, como por ejemplo al distribuir el esfuerzo de la programación en la fase de desarrollo con un porcentaje del 84%, así como en la creación del presupuesto con un porcentaje del 74%. Los métodos de estimación de costos pueden ser mejorados para apoyar diversos usos a lo largo de vida del desarrollo de software y aportar más valor a las organizaciones.

Por otra parte las empresas hacen estimación de costos al establecer los requisitos del proyecto (74%), seguido del estudio de factibilidad (67%). Estos resultados son razonables ya que en el

análisis de viabilidad del proyecto y presupuesto se necesitan los principios de estimación. Existen muchos riesgos en las etapas iniciales de desarrollo, lo que sugiere que las estimaciones sean más complejas por lo que es importante que se resuelva esta problemática al tener en cuenta cada uno de las variables que intervienen en el proyecto y de esta manera proporcionar un mejor soporte para la estimación temprana.

Del resultado de este estudio es importante contextualizar en el sector cuales son las barreras o dificultades al aplicar los métodos de estimación de costos de software con el fin de generar estrategias para mejorar su aplicación. Se evidencia que los modelos de estimación cuestan mucho esfuerzo para recopilar datos, configurar parámetros y calibrar modelos (58%), la organización no cuenta con suficiente inversión para mejorar las estimaciones (45%). Por lo anterior es importante realizar un diagnóstico de como las empresas realizan sus estimaciones y si las causan que consideran que generan barreras al usar los métodos están alineadas con lo consultado del estudio realizado en China.

Con un estudio similar realizado en Noruega, los autores Moløkken-Østvold et al. (2004) sobre “la estimación de software en la Industria Noruega” proporcionan una visión general de los diferentes métodos de estimación de costos que aplican las empresas de Ingeniería de Software en sus proyectos. Se realizó un estudio a profundidad donde se recopiló información a través de 18 entrevistas estructuradas a altos directivos y 52 líderes de proyectos de software. El objetivo fue analizar los diferentes métodos de estimación, la precisión del esfuerzo y cómo afecta el sesgo en las estimaciones. Los resultados se compararon, al igual que en china con encuestas aplicadas en los años 80 y principio de los 90, en donde los tipos de hardware, los lenguajes de programación, los procesos de desarrollo, clientes y compañías de software han cambiado desde entonces.

Un aspecto a tener en cuenta en este tipo de estudio, es la frecuencia de exceso de estimación, o la precisión de la estimación promedio. Las encuestas mostraron que la mayoría de los proyectos se completaron con más esfuerzo y horario que el estimado, con un porcentaje aproximado del 30% a 40%. Las estimaciones a estudiar se dividieron en 3 categorías: Juicio experto, métodos basados en modelos como COCOMO, estimación por casos de uso, métricas FPA o métodos algorítmicos y otros.

Este estudio se basó en responder las siguientes preguntas relacionadas con la estimación de software:

RQ1: ¿Cuál es la frecuencia y magnitud del exceso de estimación de esfuerzo?

RQ2: ¿Cuál es la frecuencia y magnitud de programar con excesos de estimación?

RQ3: ¿El tamaño del proyecto afecta la exactitud de esfuerzo o sesgo?

RQ4: ¿Cuáles son los métodos de estimación (Juicio experto, estimaciones basadas en modelo o combinaciones) más utilizados en la industria?

RQ5: ¿La elección del método de estimación afecta la tendencia o precisión del esfuerzo?

RQ6: ¿Existen diferencias en la precisión o tendencia de la estimación entre las empresas que desarrollan proyectos internamente y los que lo desarrollan para los clientes?

RQ7: ¿Cómo perciben los altos directivos de las empresas el nivel de habilidad de la estimación?  
RQ8: ¿Sobre qué base selecciona un método de estimación?  
RQ9: ¿Cómo las empresas perciben la estimación en comparación con otros aspectos de desarrollo?

A fin de garantizar una muestra representativa y estratificada se realizó muestreo al azar en diferentes tipos de organización; tales como casas de software, empresas de productos de desarrollo de software, contratistas que desarrollan para los clientes, etcétera. Las empresas se clasificaron en diferentes estratos basados en los ingresos y el número de empleados. Es fundamental anotar que en el momento de la encuesta el mercado de desarrollo de software fue muy competitivo y sufrieron pérdida; ante esto, los altos directivos de las empresas redujeron sus tarifas por hora (para el desarrollo de clientes) en lugar de subestimar proyectos, siendo un factor que no tiene nada que ver con la habilidad de la estimación.

Las preguntas para esta investigación se estructuran tanto para los altos directivos a nivel empresarial (se reflejaron actitudes, creencias y orden de preferencia), como para los directores de proyectos basados en los 52 proyectos específicos. De estos proyectos, 8 se descartaron por tratarse de menos de 100 horas-hombre o porque el trabajo de desarrollo fue realizado por consultores externos y los gerentes no mantenían un registro preciso de la estimación de esfuerzo y/o real. De los 4 proyectos cuya inclusión fue significativa, dos (5%) fueron abortados sin ser contemplados en este estudio, y cinco (11%) se completaron a tiempo estimándose el esfuerzo y la funcionalidad requerida. El resto (84%) fueron desafiados con respecto al exceso de esfuerzo, excesos de horario, funciones o una combinación de estas. De los 42 proyectos que se completaron, 32 (76%) tuvieron exceso de esfuerzo, 2 proyectos (5%) terminaron sin inconvenientes y 8 (19%) presentaron menos esfuerzo que el estimado. En cuanto a horario, 26 proyectos (62%) tenían excesos, mientras que 15 (36%) se completaron a tiempo, y un (2%) antes de lo previsto.

En esta investigación con el fin de hallar la precisión de la estimación, relacionada con el esfuerzo y el horario se calculó el error relativo  $BRE = |x - y| / \min(x, y)$ ; donde  $x$  es la estimación actual y el valor de " $y$ " es el estimado. Igualmente se calcula el sesgo en la estimación con el fin de revelar una dirección de una eventual inexactitud,  $BRE_{bias} = (x - y) / \min(x, y)$  (Jørgensen y Sjøberg, 2003)

Cuando se pregunto acerca de los tipos de enfoques de estimación utilizados en los proyectos de software, 13 de las empresas respondieron que dependía cien por ciento de la estimación de expertos, 3 de utilizaban una combinación de juicio experto y modelos de estimación en el 100% del tiempo; mientras que 2 compañías utilizaban a veces juicio de experto o una combinación con modelos de estimación. De las 5 empresas que utilizaban una combinación de modelos de estimación y juicio experto, cuatro de ellas aplicaron estimación de casos de usos. Los resultados obtenidos son muy similares a estudios anteriores (ver figura 7), lo que sugiere que

los métodos de estimación en las empresas han tenido pequeñas variaciones desde la décadas de los 80.

Por otra parte se comprobó que el uso de modelos de estimación combinado con juicio experto no daba lugar a estimaciones más precisas que cuando se realizaba solo con el juicio experto. En este estudio se observó una tendencia que los proyectos más grandes son más propensos a la subestimación que los proyectos pequeños, y se demostró que los proyectos de desarrollo interno tienden a ser más precisos y menos sesgados que los proyectos relacionados externamente, donde 11 proyectos eran de desarrollo propio (interno) y 31 eran desarrollados por contrato para un cliente (externo). En este estudio se evidenció que las estimaciones en comparación con otros aspectos son consideradas “muy importante o importante”, lo cual es evidencia de la necesidad de trabajar con estimaciones en el desarrollo de proyectos de software.

Resultados Obtenidos al realizar las estimaciones	
Sobrecosto	41%
Proyecto que utilizaron más de esfuerzo del estimado	76%
Proyectos que utilizaron menos esfuerzo del estimado	19%
Exceso en la programación	25%
Proyectos terminados después del horario	62%
Proyectos terminados antes de lo previsto	2%

Figura 8. Resultados Obtenidos en el estudio. Adaptado de (Moløkken-Østvold et al., 2004)

En conclusión los datos arrojados en este estudio son similares a los reportados por otros investigadores como McAulay (1987) , Heemstra y Kusters ( 1988) y Wydenbach (1995). Se concluye que la industria de software de Noruega posee un enfoque bajo en la precisión de sus estimaciones, lo que condujo a una evaluación de las habilidades para utilizar los diferentes métodos de estimación y lograr una educación en su aplicación y producir mejoras significativas. El autor propone como trabajo futuro supervisar el desarrollo de proyectos de software en Noruega, para analizar como los procesos de mejora en esfuerzo y cambios tecnológicos pueden afectar al rendimiento de la estimación e igualmente recomienda poder aplicar este tipo de encuestas en otros países y comparar este estudio con información más actualizada.

En este estudio el 91% de los directivos respondieron que veían la estimación como un problema y solo un 9% confirmo lo contrario. Es preocupante esta cifra ya que evidencia que la manera en que se lleva a cabo estos procesos no es el mejor, al generarse horarios y esfuerzos poco



realistas, tal y como lo asegura el autor Addison (2002) al identificar estos factores de riesgos en las estimaciones. Al realizar este estudio se evidenció que las estimaciones son consideradas como un problema y a su vez como una actividad que es importante al considerar la precisión del esfuerzo en la práctica. Por lo anterior es pertinente mejorar los procesos y los esfuerzos obtenidos en la consecución de estimaciones más precisas.

### 2.5.3 Situación actual de Colombia

En la X jornada de Gerencias de proyectos TI encabezada por la Sociedad Colombiana de Ingenieros de Sistemas: ACIS, por medio de la X encuesta Nacional, dirigida por el ingeniero **Alberto Cueto Vigil** (PMP), se evidenciaron las tendencias actuales de desarrollo de proyectos Colombianos relacionados con tecnología de Informática. Este estudio fue realizado con una participación de 355 empresas participantes:

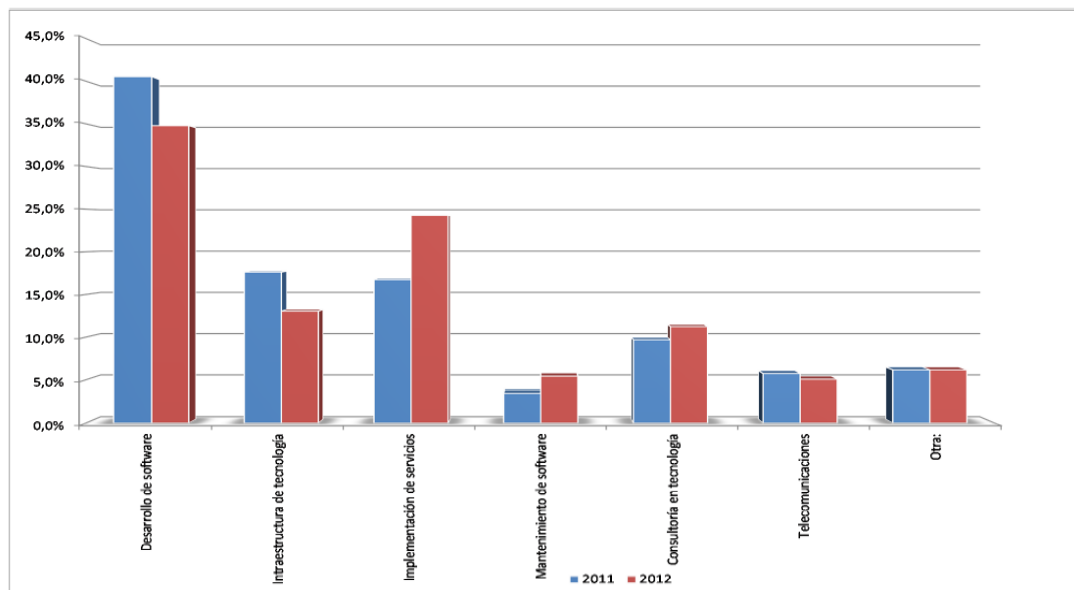


Figura 9. Tipo de proyecto de Tecnología de información en Colombia. Tomado de (Alberto Cueto Vigil, 2012)

Los proyectos de desarrollo de software tanto en el año 2012 como en el año 2011 son los que en mayor porcentaje se llevan a cabo, mostrando una diferencia significativa frente a los otros tipos de proyectos de TI.

El sector donde se desempeñan estos proyectos se pueden observar en la siguiente figura, donde el sector que mayor demanda de proyectos tiene es el sector privado, duplicando en el

2011 el sector público, y para el 2012 el sector privado triplica al sector público con un porcentaje de diferencia del 42,02%.

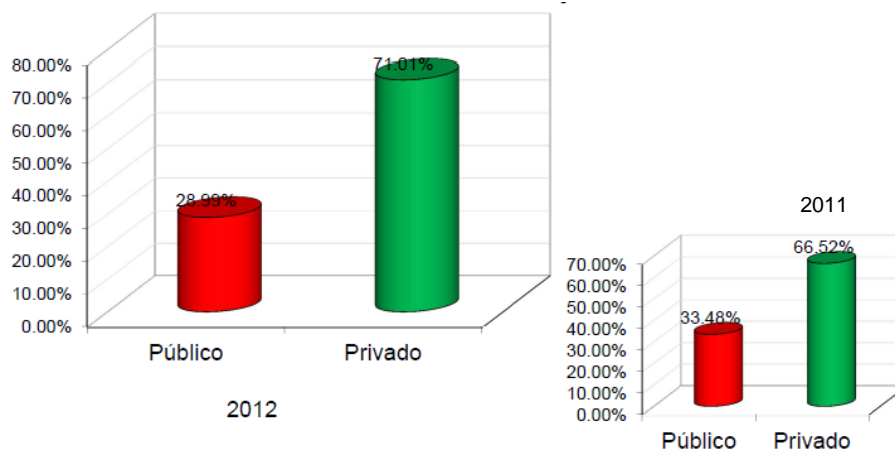


Figura 10. Distribución de sectores 2012 vs 2011. Tomado de (Alberto Cueto Vigil, 2012)

Un aspecto importante que se analizó para los proyectos de Tecnología de la Información es el presupuesto planeado vs presupuesto real, en la figura 11 correspondiente a los años 2010, 2011 y 2012 se mantiene la tendencia donde se muestra que el mayor porcentaje de proyectos establecen su presupuesto en un rango de \$100 y \$499 millones de pesos. Entre el rango de \$1.000 y \$4.999 el año del 2010 fue que obtuvo un porcentaje menor, pero en los años siguientes aumento y se considera que fue equilibrado.

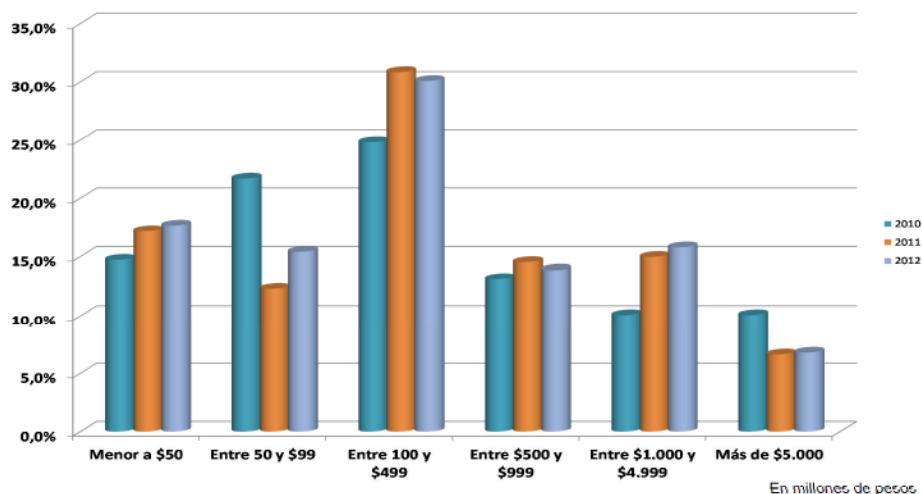
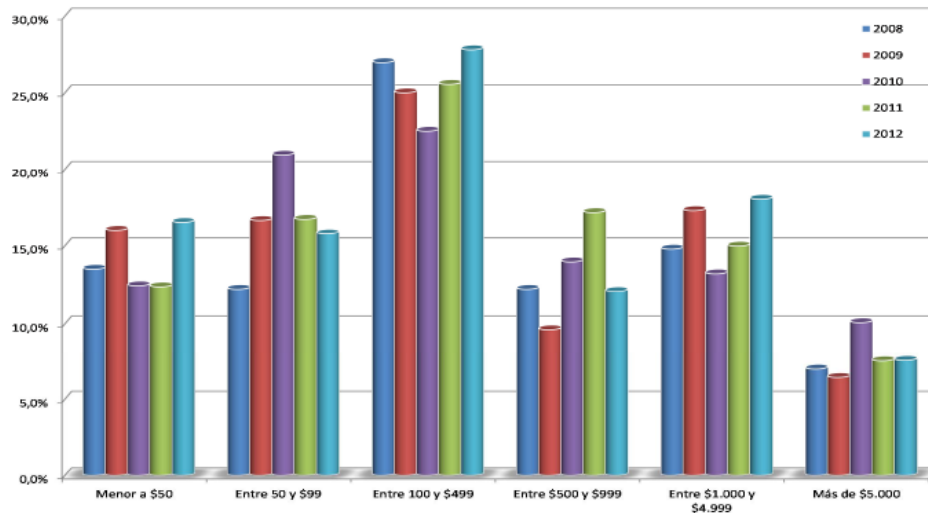


Figura 11. Presupuesto planeado de los proyectos 2010-2012 en las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012)

El presupuesto planeado con el presupuesto real se puede contrastar, evidenciándose una diferencia en los porcentajes obtenidos. En la siguiente figura se muestra la tendencia en los últimos 5 años:



**Figura 12. Presupuesto real 2008-2012 en las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012)**

Se puede observar que en el rango de \$1.000 y \$ 4.999 millones de pesos, el presupuesto real fue mayor que el planeado, en el 2012 aumento un 5%, en el 2011 se mantuvo y en el 2010 se encontraba en un 9% el presupuesto planeado y en el presupuesto real aumento a un 13%. Por otra parte, el número de proyectos que más se ejecutan que corresponden a la escala entre \$100 y \$499 millones de pesos, se muestra que en el año 2010 al ejecutarse el proyecto se gastó menos de lo presupuestado, en el año 2011 y 2012 se evidencia la misma tendencia.

En cuanto a la duración planeada se resalta en las gráficas de duración planeada y duración real un porcentaje de diferencia significativa en el rango de 13 y 15 meses. En el 2011 el porcentaje de duración planeada fue aproximadamente del 26% y la duración real obtuvo un porcentaje del 21%, en el 2012 se presentó un porcentaje de 3% de duración planeada y se obtuvo un incremento del 9% en duración real. Se puede mostrar una tendencia que a medida que el proyecto tiene un rango mayor de tiempo las duraciones planeadas y reales muestran diferencias en sus porcentajes con aumento en la duración real.

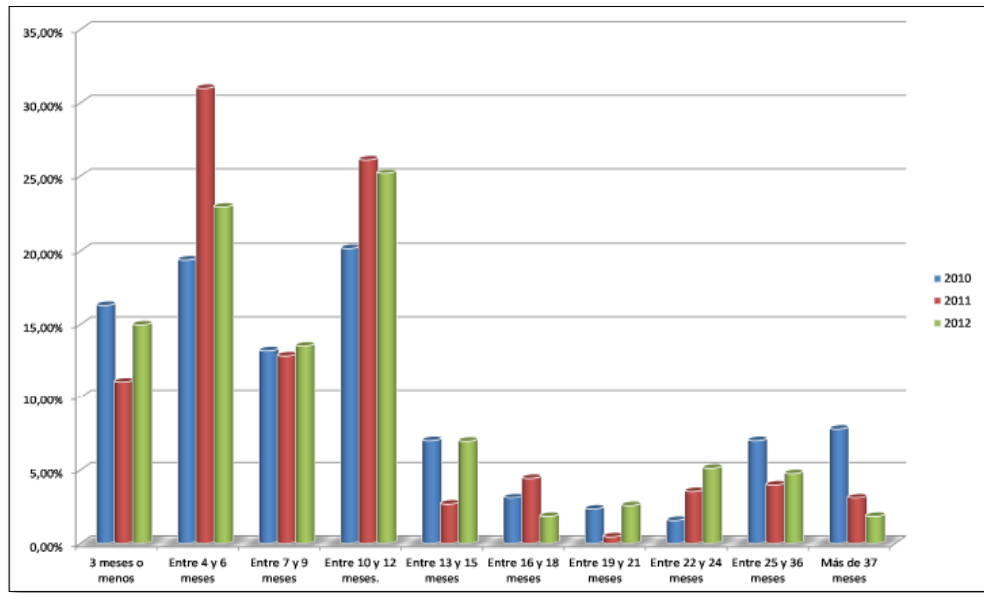


Figura 13. Duración planeada de proyectos 2010-2012 en las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012)

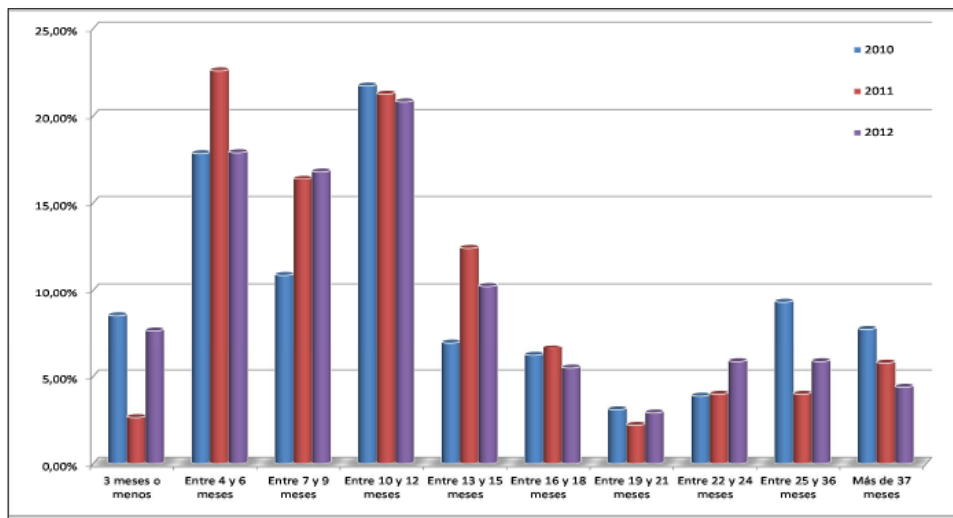


Figura 14. Duración real 2010-2012 en los proyectos de las empresas desarrolladoras de software en Colombia. Tomado de (Alberto Cueto Vigil, 2012)

Al analizar el desempeño del cronograma se puede analizar que desde el año 2010 al 2012 se muestra una tendencia similar de completar con atrasos en el cronograma, con un porcentaje del 60% de los entrevistados. En los últimos 3 años se muestra que ha bajado el porcentaje de

proyectos que se han completado con éxito ajustándose al cronograma, sin llegar arriba del 30%. Cabe resaltar que es muy pequeño los proyectos que se cancelan o abandonan.

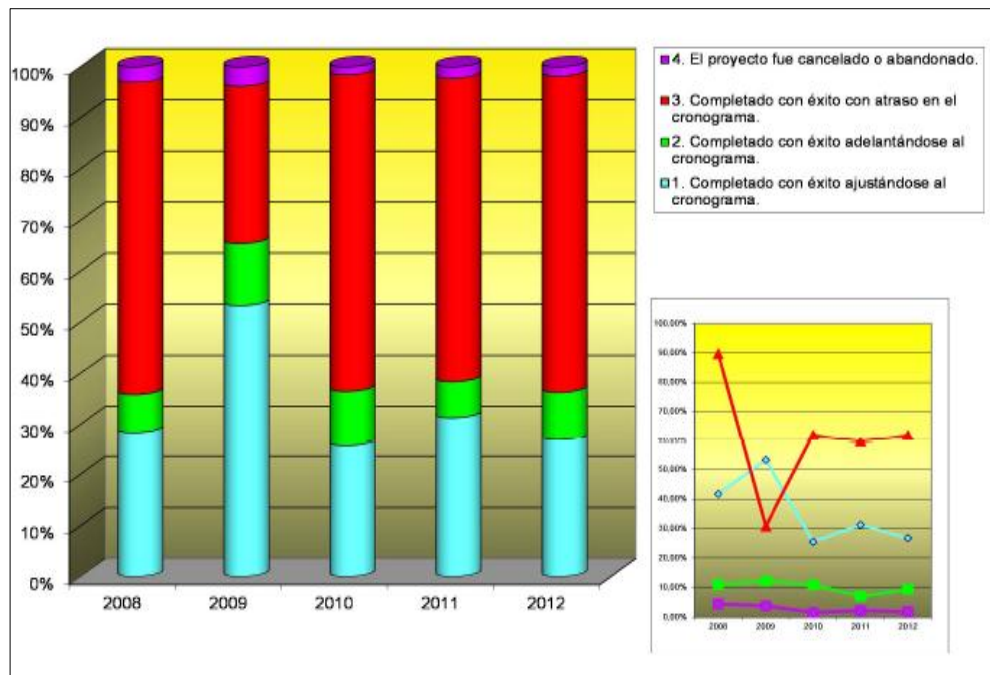


Figura 15. Desempeño en el cronograma en las empresas desarrolladoras de Software en Colombia.

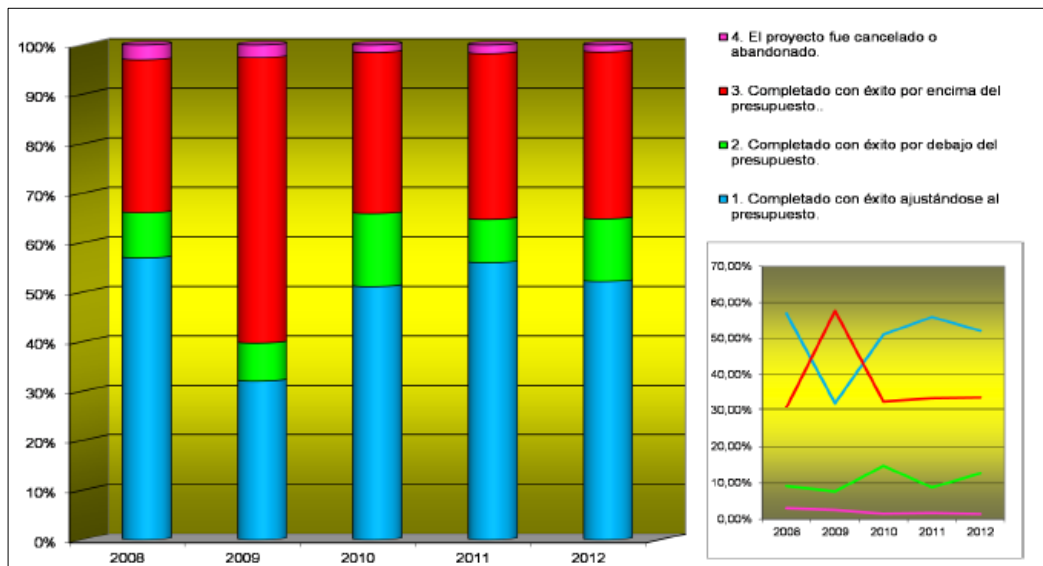


Figura 16. Desempeño del presupuesto en las empresas desarrolladoras de Software en Colombia. Tomado de (Alberto Cueto Vigil, 2012)

## 2.6 Conclusión

Hoy en día las compañías compiten por lograr un posicionamiento en el mercado, y las empresas de desarrollo de software colombianas no son ajenas a esta tendencia. Con el plan Nacional de TIC 2008-2019, el Gobierno nacional a través de las diferentes políticas del ministerio TIC, tiene el objetivo de convertir al país en una potencia de desarrollo de software a nivel mundial (Comunicaciones, 2008), lo que implica mejorar todos los procesos, como lo es la planificación en los proyectos de software que permite utilizar mejor los recursos y que cada día es más relevante dentro de la ingeniería de software (Joseba Esteban López, 2008; Bozhikova y Stoeva, 2010). Las empresas al tener procesos de estimación organizados y realistas incrementan los niveles de competitividad, generando productos de software a tiempo y de alta calidad.

Por otra parte, el software es uno de los componentes más complejo y costoso de fabricar en los sistemas de información que se utilizan actualmente. Para sistemas complejos y personalizados, un error en la estimación de costo puede marcar la diferencia entre beneficios y pérdidas para una compañía (Sommerville, 2005), ya que una de las características que debe tener un producto de software es que su costo sea adecuado, de lo contrario el proyecto puede fracasar.

De esta manera, estimar ayuda a las organizaciones a tomar mejores decisiones del negocio, ya que a partir de una correcta estimación se puede inferir la viabilidad, así como también las diferentes restricciones que se presentan en un proyecto como son: el costo, el tiempo y los recursos disponibles, estos últimos son efectos directos de los dos primeros.

Para las empresas desarrolladoras de producto de software subestimar los costos al desarrollar y realizar una mala programación puede generar un impacto perjudicial sobre la funcionalidad y la calidad del producto de software y por ende se puede ver comprometida la reputación y competitividad del desarrollador, y en situaciones extremas puede causar abandonar el proyecto. En contraste, Sobrestimar los costos y el calendario de un proyecto puede resultar una pérdida de recursos debido a la asignación redundante o perder una licitación o un contrato del desarrollo de un proyecto (Shepperd, May 2007). De esta manera, el 53% de los productos de Software no se entregan a tiempo ni dentro del presupuesto inicial, mientras el 18% de los proyectos son abandonados por completo, por lo que es importante enfatizar en la importancia del esfuerzo para predecir la exactitud del desarrollo de software. Cabe anotar que en los últimos 20 años son pocas las mejoras en la precisiones de estimaciones de costos de software, y se cree que una de las razones es la imprecisión el uso de la terminología para la estimación de esfuerzo (Group, 2004)&(Grimstad et al., 2006).

La finalidad de la estimación no es predecir el resultado del proyecto, sino determinar si los objetivos son suficientemente realistas para permitir que el proyecto sea controlado (Mario Piattini V, 2008). En la actualidad existen muchos métodos, como estimación por analogía, juicio experto, precio para ganar, Cocomo II, estimación por componentes que son utilizados para realizar estimaciones de costos y no se puede afirmar cual es la mejor debido a que se necesita de personal con experiencia en estimación, uso de información histórica, y adaptar el proyecto al

método de estimación de costos que se va a utilizar. Esto depende en gran medida, en que las empresas conozcan a profundidad su funcionamiento y sepan eliminar aquellas variables que generan estimaciones imprecisas.

Es por esto que se propone desarrollar un modelo de estimación de costos de software, con el propósito de mejorar la competitividad en las empresas desarrolladoras de software y responder adecuadamente a las exigencias del mercado actual.

## **CAPITULO III**

### **3. METODOLOGÍA PROPUESTA PARA EL MODELO DE ESTIMACIÓN DE COSTOS DE SOFTWARE**

#### **3.1. Introducción**

El método Delphi tiene como objetivo identificar un consenso por medio de un grupo de expertos en una temática específica; en este caso en los procesos para desarrollar estimaciones de costos de software.

En este capítulo se explica la importancia del método Delphi, las fases que lo componen, porque se eligió para este estudio, existiendo otras opciones de métodos cualitativos o subjetivos.

#### **3.2 Método Delphi**

El método Delphi es un nombre que proviene de la antigua Grecia del Oráculo de Delfos. El oráculo era un profeta venerado y predecía el futuro. En los tiempos modernos, la Corporación Rand utilizó por primera vez el método Delphi, después de la Segunda Guerra Mundial para analizar los probables objetivos en los Estados Unidos de bombarderos enemigos en caso de otra guerra. Debido a las opiniones dispares de los expertos militares, el método Delphi se utilizó para crear una opinión de consenso a este problema complejo.

Delphi es definido por Kavantzias, et al. (2004) como un método de estructuración de un proceso de comunicación grupal que es efectivo a la hora de permitir a un grupo de individuos, como un todo, tratar un problema complejo.

El método consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes.

Por lo tanto, la capacidad de predicción del método se basa en la utilización sistemática de un juicio intuitivo emitido por un grupo de expertos. Es decir, el método Delphi procede por medio de la interrogación a expertos con la ayuda de cuestionarios sucesivos, a fin de poner de manifiesto convergencias de opiniones y deducir eventuales consensos. La encuesta se lleva a cabo de una manera anónima (actualmente es habitual realizarla haciendo uso del correo electrónico o mediante cuestionarios web establecidos al efecto) para evitar los efectos de "líderes". El objetivo de los cuestionarios sucesivos, es "disminuir el espacio intercuartil precisando la mediana".



El proceso debe proporcionar algún tipo de retroalimentación, una valoración de la sentencia de grupo, oportunidades para que las personas revisen sus puntos de vista y un cierto grado de anonimato de los encuestados individuales. Las preguntas se refieren, por ejemplo, a las probabilidades de realización de hipótesis o de acontecimientos con relación al tema de estudio (que en nuestro caso sería el desarrollo futuro del sector que estamos analizando). La calidad de los resultados depende, sobre todo, del cuidado que se ponga en la elaboración del cuestionario y en la elección de los expertos consultados.

Por lo tanto, en su conjunto el método Delphi permitirá prever las transformaciones más importantes que puedan producirse en el fenómeno analizado en el transcurso de los próximos años.

El método Delphi es generalmente asociado a los pronósticos y planificaciones de los problemas, pero los autores de Kavantzias et al. (2004) Gregory J. Skulmoski (2000) (Gordon Xu, 2006) identificaron diferentes usos para este método incluyendo el análisis de datos históricos. Sin embargo, este estudio de estimación de Costos de Software en las empresas desarrolladoras utilizan el método Delphi para obtener una opinión de consensos de problemas futuros.

Según los autores (Kavantzias et al., 2004) un problema debe exhibir una o más de una serie de propiedades particulares para justificar el uso del método Delphi. De lo contrario los resultados pueden ser producidos de manera más precisa y exacta en un menor tiempo. Las propiedades más relevantes para este caso son:

- ✓ No hay técnicas analíticas precisas apropiadas para el problema, pero sería beneficioso tener el juicio subjetivo de un grupo de expertos.
- ✓ Los expertos requeridos para la diversidad de la problemática no tienen relación previa y tienen una gran experiencia en el área.
- ✓ Un gran número de expertos es requerido para la resolución de la problemática, convirtiendo las reuniones presenciales en algo ineficiente.
- ✓ El tiempo y el costo para organizar múltiples reuniones grupales es demasiado alto.
- ✓ Las diferencias entre los expertos deben ser preservadas para poder obtener un conjunto válido de resultados.

### **3.3 El método Delphi aplicado al proyecto de Investigación**

El método Delphi tiene como objetivo por medio de un conjunto de rondas con un panel de expertos identificar un consenso general de un tema específico. Al poner en práctica el método, la primera ronda por medio de una lista de sugerencias del tema, incentiva a los encuestados a pensar en su experiencia y conocimiento sobre los diferentes métodos de estimación de costos de software. La segunda ronda considera lo «más importante» y «lo más probable que ocurra» temas o tendencias. El objetivo de la tercera ronda del proceso es generar un consenso, donde

se envían los resultados y los expertos tienen la opción de cambiar algún punto de vista al conocer el resultado global, pero manteniendo en anonimato a los expertos.

El objetivo de cada ronda se formuló a través del "método convencional de Delphi de Linstone y Turoff (1975) como guía los autores (Kavantzias, et al., 2004) (Skulmoski, et al., 2000) (Xu y Gutiérrez, 2006 ). El diseño del formato es texto simple y utiliza un formato sencillo para incentivar a su diligenciamiento.

La primera ronda del proceso de Delphi distribuye una lista de los procesos para realizar estimaciones de Costos de software. Esta lista de temas proviene de la revisión de la literatura sobre el tema. Se presentaron las siguientes 4 categorías generales:

- ✓ **Juicio Experto.**
- ✓ **Estimación por analogía.**
- ✓ **Precio para ganar.**
- ✓ **Estimaciones por medio del uso de aplicaciones sistematizadas.**

### **3.4 Descripción del Método Delphi**

Según los autores Kavantzias et al. (2004), Skulmoski et al. (2000), Xu y Gutiérrez (2006 ) los pasos para aplicar el método Delphi son:

**Fase 1:** formulación del problema: Se trata de una etapa fundamental en la realización del método. La importancia de esta fase es definir con precisión el campo de investigación es muy grande por cuanto que es preciso estar muy seguros de que los expertos reclutados y consultados poseen todos la misma noción de este campo.

La elaboración del cuestionario debe ser llevada a cabo según ciertas reglas: las preguntas deben ser precisas, cuantificables (tratan por ejemplo sobre probabilidades de realización de hipótesis y/o acontecimientos, la mayoría de las veces sobre datos de realización de acontecimientos) e independientes (la supuesta realización de una de las cuestiones en una fecha determinada no influye sobre la realización de alguna otra cuestión).

**Fase 2:** elección de expertos: La etapa es importante en cuanto que el término de "experto" es ambiguo. Con independencia de sus títulos, su función o su nivel jerárquico, el experto será elegido por su capacidad de encarar el futuro y posea conocimientos sobre el tema consultado.

La falta de independencia de los expertos puede constituir un inconveniente; por esta razón los expertos son aislados y sus opiniones son recogidas por vía postal o electrónica y de forma anónima; así pues se obtiene la opinión real de cada experto y no la opinión más o menos falseada por un proceso de grupo (se trata de eliminar el efecto de los líderes).

**Fase 3:** Elaboración y lanzamiento de los cuestionarios (en paralelo con la fase 2): Los cuestionarios se elaborarán de manera que faciliten, en la medida en que una investigación de estas características lo permite, la respuesta por parte de los consultados. Preferentemente las respuestas habrán de poder ser cuantificadas y ponderadas (año de realización de un evento, probabilidad de realización de una hipótesis, valor que alcanzará en el futuro una variable o evento).

Se formularán cuestiones relativas al grado de ocurrencia (probabilidad) y de importancia (prioridad), la fecha de realización de determinados eventos relacionadas con el objeto de estudio: necesidades de información del entorno, gestión de la información del entorno, evolución de los sistemas, evolución en los costes, transformaciones en tareas, necesidad de formación.

En ocasiones, se recurre a respuestas categorizadas (Si/No; Mucho/Medio/Poco; Muy de acuerdo/ De acuerdo/ Indiferente/ En desacuerdo/Muy en desacuerdo) y después se tratan las respuestas en términos porcentuales tratando de ubicar a la mayoría de los consultados en una categoría.

**Fase 4:** desarrollo práctico y explotación de resultados: El cuestionario es enviado a cierto número de expertos (hay que tener en cuenta las no-respuestas y abandonos).

Naturalmente el cuestionario va acompañado por una nota de presentación que precisa las finalidades, el espíritu del Delphi, así como las condiciones prácticas del desarrollo de la encuesta (plazo de respuesta, garantía de anonimato). Además, en cada cuestión, puede plantearse que el experto deba evaluar su propio nivel de competencia.

El objetivo de los cuestionarios sucesivos es disminuir la dispersión de las opiniones y precisar la opinión media consensuada. En el curso de la 2ª consulta, los expertos son informados de los resultados de la primera consulta de preguntas y deben dar una nueva respuesta y sobre todo deben justificarla en el caso de que sea fuertemente divergente con respecto al grupo. Si resulta necesaria, en el curso de la 3ª consulta se pide a cada experto comentar los argumentos de los que disienten de la mayoría. (Puertas Del Castillo, 2011)

Existen diferentes métodos cualitativos para pronósticos, y se caracterizan porque no requieren de una abierta manipulación de datos sino que hacen uso del juicio de quien pronostica. Por su naturaleza éstos suelen ser subjetivos y no utilizan modelos matemáticos. Las técnicas cualitativas se usan cuando no se tiene disponibilidad de información histórica. Entre los métodos se destacan la investigación de mercados, normalmente se usa para probar una hipótesis acerca de mercados reales. Suelen ser cuestionarios estructurados que se envían a los clientes potenciales del mercado solicitando en ellos opinión acerca de productos o productos potenciales e intentan muchas veces averiguar la probabilidad de que los consumidores demanden ciertos productos o servicios. Por otra parte el método cualitativo de analogía de ciclo de vida se usa cuando se desea lanzar un nuevo producto y se tiene en cuenta el marco de tiempo, duración y madurez del producto, velocidad de crecimiento y decadencia,

tamaño de la demanda global sobre todo en la fase de madurez. El juicio informado es uno de los menos confiables al utilizar el pronóstico como una oportunidad de metas optimistas. Miedo de establecer un rango menor sobre una escala de consenso. El método Delphi según se ha explicado en este capítulo se utiliza para pronósticos a largo plazo y se basa en la experiencia de expertos por lo que es uno de los métodos más confiables tipo cualitativo. En el siguiente cuadro se analizan los métodos según el horizonte de beneficio. (Reich, 2009)

<b>Método cualitativo</b>	<b>Horizonte de beneficio</b>
Método Delphi	Mediano y largo plazo
Juicio informado	Corto plazo
Analogía de ciclo de vida	Mediano y largo plazo
Investigación de Mercado	Corto y mediano plazo

Figura 17. Métodos Cualitativos.

### 3.5 Problemas Comunes del Método Delphi

Linstone y Turoff (1975) Enumeran ocho problemas básicos que normalmente se asocian con el uso del método Delphi. Esta sección se analiza brevemente cada uno de estos inconvenientes y el diseño de este particular proceso de Delphi para minimizar el impacto de los inconvenientes.

La mayoría de la gente tiene un horizonte de planificación a corto plazo, por lo que a la hora de prever el futuro hay una tendencia a descartar el futuro. Por lo tanto, cuanto más se indaga sobre el futuro más conservadoras se vuelven las predicciones.

Para mitigar esta tendencia los escenarios utilizados en la segunda ronda del proceso varía el período de predicción.

Un impulso a la simplificación excesiva es otro error común. Esto puede conducir a muchos diferentes problemas en el proceso, uno de ellos es la consideración de una pequeña parte de una compleja interacción, y la consecuente pérdida de cualquier análisis. Este efecto es superado en parte, por la extensa revisión bibliográfica antes del inicio del proceso, pero no se puede descartar por completo, ya que sólo una cantidad limitada de información se distribuye en el panel.

Los panelistas pueden tener una experiencia en el ámbito de la investigación, debido a su enfoque singular. Por una compleja interacción de impulsos los expertos de diferentes áreas están obligados a proporcionar predicciones más exactas. En este informe, aunque el panel es relativamente pequeño, hay una amplia gama de personas de tecnología, negocios y antecedentes académicos.

La aplicación del proceso también puede tener un impacto sobre la calidad del resultado, ya sea por culpa del coordinador. Este proceso Delphi se ha diseñado tras consultar Linstone y Turoff (1975). Además, un piloto de cada ronda se llevó a cabo para descubrir los problemas asociados con la redacción o el lenguaje de la información.

Un problema común, especialmente para la predicción tecnológica, es un exceso de optimismo para la previsión a corto plazo. Este se caracteriza por una subestimación de la hora de traer un nuevo sistema de mercado, haciendo caso omiso de las complejidades de la investigación. Del mismo modo, hay una tendencia de pesimismo para predicciones a largo plazo debido a nuestra experiencia hasta la fecha, y por lo tanto, una falta de reconocimiento o permitir cualquier método totalmente nuevo.

La aplicación inadecuada de método Delphi es una falta de entusiasmo por el uso. El uso del proceso para este estudio se ha explicado con anterioridad en este capítulo. Por último, el proceso Delphi no se puede utilizar engañosamente ni manipular los resultados.

Cada uno de estas dificultades, descrito previamente, estará presente en cualquier estudio a un mayor o menor grado según (Linstone y Turoff, 1975). Por lo tanto, un nuevo análisis de las limitaciones de este estudio se presenta en la sección de Resultados. (Puertas Del Castillo, 2011)

### **Número óptimo de expertos**

Aunque no hay forma de determinar el número óptimo de expertos para participar en una encuesta Delphi, estudios realizados por investigadores de la Rand Corporation Dalkey, et al. (1999) señalan que si bien parece necesario un mínimo de siete expertos habida cuenta que el error disminuye notablemente por cada experto añadido hasta llegar a los siete expertos, no es aconsejable recurrir a más de 30 expertos, pues la mejora en la previsión es muy pequeña y normalmente el incremento en coste y trabajo de investigación no compensa la mejora. (Puertas Del Castillo, 2011)

## **3.6 Conclusiones**

Al identificar las tendencias o escenarios de como las empresas pueden realizar sus procesos de estimación de costos, el método Delphi es una opción viable ya que por medio de la ayuda de sucesivos cuestionarios sirve para deducir eventuales consensos. A pesar que se considera un método cualitativo o subjetivo proporciona una visión a mediano y largo plazo, siendo fundamental su aplicación en este estudio para conocer la media de cuáles son los escenarios de estimación con mejor clasificación y valorados.

## CAPITULO IV

### 4. RESULTADOS DE LA INVESTIGACIÓN

#### 4.1 Introducción

En este capítulo se detalla el proceso utilizado para aplicar el método Delphi con el panel de expertos. Se detallan cada una de las fases ejecutadas y el resultado obtenido. Se realiza un análisis de las respuestas obtenidas por escenarios incluyendo el coeficiente de variación y desviación estándar. Se utilizó la herramienta Statgraphics centurium XV.II para estudiar el factor de clasificación y valoración sobre los procesos de estimación de costos de software en los escenarios que se plantearon con el método Delphi.

Igualmente en este capítulo se presentan recomendaciones generales, sugerencias y limitantes del proyecto

#### 4.2 Proceso de Investigación

El grupo de Investigación de Ingeniería de Sistemas de la Corporación Universidad de la Costa CUC viene desarrollando proyectos en conjunto con empresas pertenecientes al clúster Caribe TIC Barranquilla, por lo cual fue de interés validar el modelo con este grupo de empresas desarrolladoras de software. Los expertos de las diferentes empresas fueron seleccionados con la ayuda de una base de datos proporcionada por la cámara de comercio de Barranquilla y el clúster Caribe TIC Barranquilla, que reúne a las empresas desarrolladoras de software de la ciudad. Confirmaron su apoyo 16 expertos, pero solo 10 diligenciaron sus comentarios y observaciones en cada una de las rondas, la cual tuvo una duración de 3 meses

En cada ronda se utilizó como medio de comunicación el correo electrónico donde se les explicó a los expertos por medio de instrucciones detalladas en qué consistía la metodología y como anexo se adjuntó un archivo en Word para su diligenciamiento (ver anexo B). A continuación se detalla por cada una de las rondas el número de envíos y replicas:

Tabla 2. Cronograma de envío de Encuestas

No de ronda	Fecha de envío	Fecha de plazo para envío	Fecha de recordatorio	Total de envíos	Total de replicas
1	3 Abril	18 de Abril	15 de Abril	16	5
2	22 Mayo	6 de Junio	1 de Junio	10	10

3	20 de Junio	15 de Junio	12 de Junio	10	10
---	-------------	-------------	-------------	----	----

### 4.3 Primera Ronda - La Formulación de los Escenarios

Los panelistas analizaban los escenarios propuestos y realizaban comentarios según la experiencia en los procesos de estimación de costos de software desde su compañía. Los objetivos planteados para esta primera ronda fueron estimular a los panelistas o expertos sobre cuáles son las variables que afectan sobre estimaciones más precisas, e identificar cambios o recomendaciones sobre los escenarios planeados.

Los escenarios se formularon al consultar la revisión bibliográfica sobre estudios a nivel mundial de cuáles son los métodos de estimación más usados según (Da Yang et al., 2008) & (Moløkken-Østfold et al., 2004):

Año	(McAulay, 1987)	1989(Heemstra, 1990)	(Wydenbach, 1995)	(Moløkken-Østfold et al., 2004)	2007 (Nuevo)
Porcentaje utilizado para cada uno de los métodos(más de un posible método)					
Consultor experto		26%	86%	100%	70%
Intuición y experiencia	85%	62%			
Analogía		61%	65%	28%	70%
Modelo de costos de Software	13%	14%	26%		15%
Precio para ganar		8%	16%		53%
Capacidad Relativa		21%	11%		
Top Down			13%		
Bottom-up			51%		
Otros	12%	9%	0%		

De la revisión bibliográfica se distribuyeron una lista de 4 escenarios iniciales, los cuales son el resultado de los métodos más utilizados según el referente de estudios a nivel mundial.

En el anexo B se listan la descripción de los escenarios para ser analizados por los panelista. No se describe a que método corresponde cada escenario propuesto, ya que el método Delphi recomienda no especificarlo con el fin de no generar sesgo en la etapa de reconocimiento e identificación de características.

De esta manera los escenarios propuestos corresponden a los siguientes métodos (ver Anexo B):

**Escenario A:** Juicio experto

**Escenario B:** Estimación por analogía

**Escenario C:** Precio para ganar

**Escenario D:** Manejo de aplicación para realizar estimaciones.

Al distribuir las encuestas a los 10 panelistas que se mostraron interesados en apoyar el estudio, se realizaron recomendaciones y sugirieron crearon nuevos escenarios que se acomodaban a las experiencias de como manejaban los procesos de estimación de costos en su compañía. Estos comentarios y la presentación de nuevas características fueron fundamentales para el refinamiento de los nuevos escenarios para la siguiente ronda. A continuación se detallan las observaciones presentadas por el conjunto de expertos sobre cada uno de los escenarios propuestos.



**Tabla 3: Recomendaciones y/o puntos de vista de los expertos sobre los escenarios de estimación de costos de software propuestos- Ronda 1**

Escenarios					
Expertos	A	B	C	D	OBSERVACIONES
1	Realizan estimaciones por medio de la experiencia, según la metodología. El grupo de desarrollo se reúne y proponen un límite de tiempo para desarrollar una tarea. El riesgo de realizar estimaciones de esta manera es que muchas veces los programadores se equivocan en el tiempo estimado.	No aplica, manejan una base de datos histórica (Base de datos de conocimiento), es una guía para manejar estimaciones futuras, pero hasta la fecha se está empezando a implementar. Desconocemos la fórmula propuesta.	No está completamente en función al ppto del cliente, sino de acuerdo a lo que el cliente está de acuerdo a pagar, se desarrollan las funcionalidades del producto. No es muy recomendable su aplicación, ya muchas veces que el producto a entregar no es el deseado por el cliente.	No aplica, la empresa maneja redmind para gestión de proyectos de software. Se realizan estimaciones de tiempos.	
2	No aplica en nuestra empresa. No lo utilizaríamos este método	Si utilizamos base de datos histórica, pero es necesario actualizarla. La fórmula nos parece confusa	Para no perder un cliente a veces se negocia, pero es importante no realizar promesas de tiempo.	Utilizamos herramientas de gestión para apoyarnos en las estimaciones	
3	Este escenario es utilizado en la empresa al aplicar la experiencia para identificar el tiempo para desarrollar una actividad. Normalmente el desarrollador Senior realiza esta estimación. No se realiza en conjunto sino que al asignar las actividades a los responsables del proyecto, según su experiencia identifican la duración para culminarla.	Se considera importante tener datos históricos para realizar estimaciones más precisas, sin embargo no conozco esta fórmula.	Este escenario es viable cuando el cliente tiene un presupuesto establecido, se negocia con el cliente pero sin incurrir en compromisos insostenibles. Al cerrar la negociación según lo que el cliente esté disponible a pagar se establece el alcance del software y las funciones que se incorporaran.	la empresa utiliza aplicaciones como dashable para disminuir la incertidumbre y estimar las tareas. Cabe resaltar que esta aplicación fue desarrollada por koombea.	Proponen un nuevo escenario que se tiene en cuenta para afinar los escenarios propuestos
4	Me parece un método poco acertado debido a que puede tener muchas opiniones y lo que van a suceder es que se van a ir con el costo que muestre el consenso, que no es lo mismo que el costo adecuado.	Comparto el hecho de soportarse en la base histórica de proyectos realizados, para medir el costo del nuevo proyecto. En cada estimación lo importante es disminuir el factor de incertidumbre, es decir si podemos tener certeza de los desarrollos que hay que realizar y su complejidad podemos estimar las horas ingeniero que se requieren y los tiempos de cada etapa, lo que va a dar un alto nivel de precisión en el costo. Sobre la formula expresada no estoy familiarizado con esta.	No comparto este escenario, bajo ningún punto de vista. Cada desarrollo tiene un costo y debe dejarle utilidad a la empresa que lo desarrolla, de lo contrario será un desastre.	Esta metodología puede ser de gran ayuda, pero volvemos a la base del proyecto y es eliminar el nivel de incertidumbre para poder estimar el esfuerzo real de cada tarea. La herramienta provee una metodología organizada y probada pero si las estimaciones de los esfuerzos no son las correctas debido al grado de incertidumbre, el resultado será igualmente incorrecto.	
5	Estoy de acuerdo, la empresa utiliza el enfoque de planning Poker, que reúne al grupo de desarrolladores que van a participar en el proyecto.	Al tener el problema de rotación de personal esto hace que las estadísticas se vuelvan inválidas rápidamente. La empresa tiende a ser muy dinámica y se vincula nuevo personal a diferentes proyectos. Es interesante este método.	La empresa costea dependiendo del nivel económico del cliente. Hemos tenido la experiencia con este escenario. Pero no es recomendable y actualmente no lo utilizamos	No es recomendable su uso, pero si apoyarse en herramientas de gestión de proyectos de software	

6	La empresa calcula el esfuerzo de esta manera, pero utilizando la llamada baraja de cartas para llegar a un consenso. Puede variar el tiempo estimado dependiendo del perfil Ing. senior, junior, coordinadores. Las personas expertas son consideradas los mismos desarrolladores asignados a un proyecto.	No manejan este escenario. Es interesante, pero consideramos que cada proyecto es diferente, sin embargo es importante tener una base de conocimientos para almacenar el histórico de las estimaciones y utilizarlas como referentes	Consideramos que es un escenario poco viable para la empresa, no es recomendado. En ningún momento han tenido que manejar la operación de esta manera	No utilizamos este escenario, pero utilizamos la herramienta de gestión de proyectos Doproject	
7	Inconvenientes con Expertos dentro y fuera dispuestos y en los tiempos requeridos para la estimación	No es confiable por haber factores humanos que afectan en un proyecto en menor o mayor medida por tanto la métrica fallaría	Escenario no viable, ya que en un momento se generaran pérdidas en la compañía al igual que el cliente, esto se producen en clientes que no tienen la madurez informática, es decir conocimientos mínimos de como es el proceso de desarrollo de Software, su complejidad y sus diferencias de la producción industrial de otros bienes y servicios de la sociedad, que no son aplicables al desarrollo de software.	Hoy día el número de líneas de código es irrelevante, la mayoría de IDEs generan código que distorsiona el costo.	Proponen un nuevo escenario que se tiene en cuenta para afinar los escenarios propuestos
8	Solo usamos esto cuando necesitamos un estimado a corto plazo. Y normalmente no lo consideramos como el estimado final. . Manejamos planning Poker	Esto daría un mejor estimado, pero teniendo en cuenta el programador que se usaría para estos estimados. Es importante llevar un histórico de su desempeño.	Eso no lo recomiendo para nadie.	No lo utilizamos, pero manejamos herramientas de gestión	
9	La empresa utiliza Planning Póker, que es una técnica para calcular una estimación basada en el consenso, en su mayoría utilizada para estimar el esfuerzo o el tamaño relativo de las tareas de desarrollo de software. Es una variación del método Wideband Delphi. Es utilizado comúnmente en el desarrollo ágil de software, en particular en la metodología Extreme Programming.	No lo maneja el área de desarrollo, el área administrativa lo maneja en cuanto tiempo y costo del proyecto en general.	han tenido la experiencia. Quitar funciones del software y generar un precio. No es la mejor manera.	No utilizan ninguna herramienta.	Proponen un nuevo escenario que se tiene en cuenta para afinar los escenarios propuestos
10	Para desarrollos nuevos se basan en la experiencia que han tenido con proyectos anteriores. Buscan desarrolladores que han trabajado en experiencia en el proyecto actual, se reúnen y dan su opinión frente al tiempo y el costo del proyecto, llegando a un consenso. Se tiene en cuenta el alcance del proyecto y dependiendo de las actividades realizan las estimaciones.	Evalúan el proyecto anterior y tiene en cuenta imprevisto que no se tuvieron anteriormente en cuenta al costo y ejecución del proyecto. No conocen la fórmula	Han tenido la experiencia, pero no recomiendan este escenario. Ven un cliente potencial y para no perderlo negocian el precio. Las actualizaciones las realizan en el contrato de mantenimiento.	No utilizan ninguna herramienta.	

#### 4.4 Segunda ronda: Clasificación y valoración

Al obtener información de la primera ronda se mejoraron y eliminaron los escenarios que no estaban acordes a la realidad del entorno estudiado. Igualmente se agregaron escenarios fusionados de diferentes expertos que coincidían en sus apreciaciones y que era importante tener en cuenta.

Los objetivos de realizar esta segunda ronda fueron:

- ✓ Obtener una puntuación de la influencia de la estimación de costos de software en los escenarios o tendencias propuesta.
- ✓ Obtener una calificación en la probabilidad de ocurrencia en los diferentes escenarios.
- ✓ Analizar los resultados, para analizar áreas de acuerdo y contención.

A los expertos se les pidió en esta ronda clasificar la importancia de los escenarios con una escala de cinco puntos, siendo 1 el primero que aplicarían y 5 el último. (Ver anexo C) Igualmente los participantes tuvieron que valorar los escenarios en una escala del 1 al 5, donde 5 expresa su total acuerdo y 1 su total desacuerdo con las diferencias tendencias generándose una probabilidad de ocurrencia. A continuación se muestra las tablas que se utilizó para realizar estas mediciones:

**Tabla 4. Escala de clasificación y valoración de cada uno de los escenarios**

Clasificar este tema (Escenario A1 de estimación de costos de software)					De 5
<b>Valoración de este tema :</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Resaltar la puntuación	Totalmente de acuerdo	Parcialmente de acuerdo	Ni en acuerdo/ ni en desacuerdo	Parcialmente en desacuerdo	Totalmente en desacuerdo
<b>Comentario o punto de vista alternativo:</b>					

#### 4.5 Tercera ronda: Confirmación de respuestas

El objetivo de esta ronda es que los expertos puedan verificar la información suministrada. El ejercicio que se realizó fue consolidar la información obtenida y enviarla a cada uno de los

panelistas con el fin de que cada uno conociera el consenso general de todo el equipo, así como los comentarios y puntos de vista generados.

En esta ronda el experto realiza ajustes de clasificación y valoración y tiene la opción de responder a comentarios realizados por otros panelistas.

#### 4.6 Calificación por escenarios

En la primera ronda los expertos realizaron sus apreciaciones sobre los escenarios propuestos. De los 4 escenarios se realizaron ajuste y recomendaciones, se agregaron nuevos escenarios e incorporaron características a los propuestos. De esta manera en la segunda ronda quedaron refinados para clasificar y evaluar 5 escenarios donde también se podían realizar comentarios o puntos de vista alternativos.

A continuación se muestran cada uno de los escenarios, la valoración y clasificación obtenida con los escenarios ya refinados (en la ronda 2):

##### Escenario 1:

Una de las formas de realizar la estimación de esfuerzo o tamaño relativo de las tareas de desarrollo se basa en utilizar una baraja de cartas que se encuentran enumeradas mostrando la secuencia Fibonacci: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100. Se utiliza esta secuencia para identificar la incertidumbre de cada una de las estimaciones. Cada uno de los integrantes del proyecto posee un mazo de cartas; el moderador que puede ir guiado por el coordinador del proyecto proporciona las características de la tarea a desarrollar, y el grupo puede intervenir para aclarar sus dudas.

Cada persona coloca su carta boca abajo, de forma simultanea que representa la estimación para una tarea. Con la serie Fibonacci se define que representa las unidades de las cartas, ya sean días, u horas.

Las personas que saquen números alejados del promedio pueden dar su justificación, y se repite el proceso hasta que se llegue a un consenso. Después de esto se sigue el mismo proceso pero con otra tarea.

Clasificación													
Escenario 1	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
		4	5	5	5	4	4	3	4	1	5	4	1,25

Valoración													
Escenario 1	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
		2	1	1	1	5	4	2	4	4	1	2,5	1,58

### Escenario 2:

La empresa considera que es importante crear una base de datos histórica para medir el costo de un nuevo proyecto. La base de conocimiento tendría información como descripción del requerimiento, complejidad del requerimiento, tiempo estimado, tiempo real, duración general del proyecto en horas, valor por hora dependiendo del perfil del desarrollador, etc.

En cada estimación lo importante es disminuir el factor de incertidumbre. Si el desarrollador tiene certeza de los desarrollos que hay que realizar y su complejidad puede estimar las horas que se requieren y los tiempos de cada etapa, lo que va a dar un alto nivel de precisión en el costo. Este tipo de información histórica genera estadísticas del tiempo para realizar una tarea y es el referente para nuevas estimaciones.

Clasificación													
Escenario 2	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. De variación
		5	4	2	1	5	5	4	2	5	2	3,5	1,58

Valoración													
Escenario 2	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
		1	3	5	5	2	1	2	5	1	5	3	1,83

### Comentario experto 3:

El ideal siempre será que se tengan en cuenta los datos históricos, el problema es que generar esa base de conocimiento y mantenerla es bien complicado y casi nunca pasa en las empresas, ya que no hay una cultura sobre esto.

**Comentario experto 5:** Considero que aunque puede tener alguna utilidad, los proyectos en nuestro caso difieren mucho unos de otros

### Comentario experto 7:

Énfasis en la negociación y no en criterios objetivos como el Alcance detallado

### Escenario 3:

La empresa debe establecer el alcance y los requisitos del proyecto en cada una de las fases para realizar una buena estimación. Por cada una de las fases se descomponen los requisitos en tareas o funcionalidades más específicas y sobre estas se realizan estimaciones de tiempo, así como el valor por hora de desarrollo dependiendo del perfil del desarrollador que la va a realizar. Se establece el tiempo de entrega con el cliente y si el cliente decide que el valor establecido es muy alto se realizan negociaciones y se eliminan funcionalidades hasta llegar a un valor en que se sienta satisfecho, esto normalmente se hace cuando la empresa cuenta con un producto que comercializa.

Clasificación													
Escenario 3	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
		3	2	3	3	2	2	2	1	2	3	2,3	0,67

Valoración

Escenario 3	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
	5	5	5	4	5	5	3	5	3	5	4,5	0,85	0,19

#### **Comentario experto 2:**

Esto daría un mejor estimado, pero teniendo en cuenta el programador que se usaría para estos estimados.

#### **Comentario experto 8:**

Nos basamos en hacer la descomposición teniendo en cuenta el número de requerimientos, tamaño de las tareas en cada requerimiento, número y complejidad de las interfaces y la prioridad por tareas

#### **Escenario 4:**

El cliente se considera con poca fundamentación para establecer requerimientos, solo cuenta con una idea general del producto que quiere. La empresa desarrolladora genera un producto mínimo viable. Se realizan entregas parciales al cliente (pueden ser semanales) el cliente lo prueba y con las recomendaciones se agregan funcionalidades y modificaciones hasta llegar al producto deseado.

En el ciclo de vida de desarrollo se cuenta con la participación activa del cliente y al desarrollo incremental con interacciones muy cortas. Una de las características es que los requisitos son cambiantes, por lo que el cliente puede ver en tiempo real las actividades que está desarrollando el grupo de desarrollo, así como el número de horas que llevará a culminarse dicha actividad. Para realizar el seguimiento de las tareas el coordinador del proyecto se reúne con su grupo de trabajo y se verifica el estado de los sprints. Es importante establecer prioridad en las tareas ya que el cliente verifica y prueba las funciones básicas, se realiza un feedback para identificar las mejoras.

**Clasificación**

Escenario 4	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
	1	1	1	2	3	1	1	5	3	1	1,9	1,37	0,72

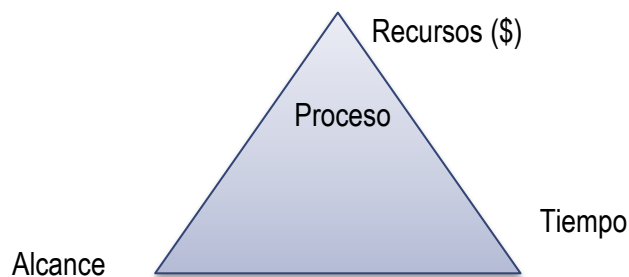
Valoración													
Escenario 4	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
	1	5	5	5	5	4	4	4	4	4	4	4,1	1,20

**Comentario experto 5:**

Solo funciona en los países desarrollados, lo veo difícil de aplicar en Colombia pero me parece muy importante.

**Escenario 5:**

Se debe tener en cuenta el alcance del proyecto para realizar una buena estimación del costo de este. En muchas ocasiones existe un problema de aprendizaje donde el líder del proyecto y desarrolladores no entiende lo que desea el cliente. Es importante primero conocer cómo opera el cliente, para lo cual se deben realizar reuniones con el cliente. Al determinar cómo se va a hacer, se determina el tiempo, el cual impacta en los recursos.





Para determinar el alcance, se tiene en cuenta los requisitos en cada una de las fases del software:

1. Análisis
2. Diseño
3. Desarrollo
4. Pruebas
5. Despliegue de entregas

Se utiliza el siguiente cuadro para estimar los costos, teniendo en cuenta que cada persona trabaja cuatro horas diarias. Se descompone en funcionalidad o actividades (con la lista anterior), siendo mucho más fácil estimar el tiempo de desarrollo del proyecto.

<b>Código</b>	<b>Funcionalidad/Actividad</b>	<b>Horas</b>
C1	Horas Efectivas	
C2	Eficiencia	
.	Horas disponibles	Dividir horas efectivas/eficiencia
.	Horas semanales	
.	Tiempo de entrega por semanas	
.	Valor hora Base	
	Costos de software	

El proyecto avanza por medio de iteraciones con el cliente que se denominan sprint, cada Sprint tiene una duración de 1 a 4 semanas.

La empresa con su mayoría de clientes firma un contrato de soporte, en el cual constantemente está realizando mantenimiento al software según las nuevas necesidades o reportes que el cliente necesite.

<b>Clasificación</b>													
<b>Escenario 5</b>	<b>Experto 1</b>	<b>Experto 2</b>	<b>Experto 3</b>	<b>Experto 4</b>	<b>Experto 5</b>	<b>Experto 6</b>	<b>Experto 7</b>	<b>Experto 8</b>	<b>Experto 9</b>	<b>Experto 10</b>	<b>Promedio</b>	<b>Desv. Estándar</b>	<b>Coef. de variación</b>
		2	3	4	4	1	3	5	3	4	4	3,3	1,16

Valoración													
Escenario 5	Experto 1	Experto 2	Experto 3	Experto 4	Experto 5	Experto 6	Experto 7	Experto 8	Experto 9	Experto 10	Promedio	Desv. Estándar	Coef. de variación
		5	2	2	3	5	3	1	4	3	4	3,2	1,32

#### 4.7 Análisis de los resultados de los factores Clasificación y Valoración

Los resultados al aplicar el método Delphi para buscar el consenso de los expertos se describen a continuación y se realiza un análisis detallado realizando comparación de medias, verificación de varianza, prueba de normalidad e independencia y distribución de normalidad.

Para analizar los datos obtenidos, se utiliza un diseño experimental de las repuestas de clasificación y valoración ofrecida por parte de los expertos. Este análisis arrojará un resultado cualitativo de las tendencias o escenarios para tomar decisiones de cuales métodos son los más significativos para los panelistas.

##### 4.7.1 Definición del problema y factores de clasificación

A continuación se muestra los resultados de clasificación y valoración sobre los 5 escenarios evaluados por los 10 expertos consultados para este estudio.

El factor que se analiza estadísticamente es la clasificación de los métodos de estimación de costos de software, en los diferentes escenarios o tendencias. A continuación se muestra los valores finales obtenidos por la clasificación

Tabla 5. Respuesta de experto factor de clasificación en los escenarios

Expertos	Escenarios				
	1	2	3	4	5
1	4	5	3	1	2
2	5	4	2	1	3
3	5	2	3	1	4

4	5	1	3	2	4
5	4	5	2	3	1
6	4	5	2	1	3
7	3	4	2	1	5
8	4	2	1	5	3
9	1	5	2	3	4
10	5	2	3	1	4

**Factor de Interés:** Clasificación sobre los métodos de estimación de costos de software.

**Niveles del factor:** 5 escenarios (Escenario A, Escenario B, Escenario C, Escenario D, Escenario E)  $K = 5$ .

**Variables de Interés:** = Clasificación de los Experto

**Numero de réplicas (Expertos):**  $n = 10$

**Numero de observaciones:** 50

### Hipótesis del problema

$\mu_1$  = Media de la clasificación de los expertos en el escenario A.

$\mu_2$  = Media de la clasificación de los expertos en el escenario B.

$\mu_3$  = Media de la clasificación de los expertos en el escenario C.

$\mu_4$  = Media de la clasificación de los expertos en el escenario D.

$\mu_5$  = Media de la clasificación de los expertos en el escenario E.

$H_0$ :  $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = 0$  ; Se rechaza  $H_0$  en función de  $H_1$

$H_1$ :  $\mu_i \neq \mu_j$  para algunos  $i, j$

La hipótesis nula se define como la igualdad de las medias de clasificación de cada uno de los expertos en los diferentes escenarios y la hipótesis alternativa se define como la diferencia de al menos dos de las medias de clasificación de los expertos en los diferentes escenarios.

En caso de no rechazar  $H_0$  se concluye que las medias de clasificación de los 5 escenarios son estadísticamente iguales; pero si se rechaza, se concluye que al menos dos de ellos son diferentes.

### Modelo Estadístico:

$$Y_i = \mu + \tau_i + \varepsilon_i$$

### **Análisis de los resultados obtenidos:**

Al analizar el resumen estadístico, se puede observar la desviación estándar en los diferentes escenarios. El escenario 3 obtuvo una clasificación promedio "2,3", con una tendencia a variar por debajo o encima de dicho valor en 0,67, siendo la menor desviación de todo el diseño experimental. Igualmente el escenario 4 obtuvo una muy buena clasificación al tener un puntaje promedio de 1,9 con una desviación estándar de 1,37.

**Tabla 6. Resumen estadístico para la clasificación**

	<b>Recuento</b>	<b>Promedio</b>	<b>Desviación Estándar</b>	<b>Coefficiente de Variación</b>
<b>Escenario 1</b>	10	4,0	1,24722	31,1805%
<b>Escenario 2</b>	10	3,5	1,58114	45,1754%
<b>Escenario 3</b>	10	2,3	0,674949	29,3456%
<b>Escenario 4</b>	10	1,9	1,37032	72,1221%
<b>Escenario 5</b>	10	3,3	1,1595	35,1364%
<b>Total</b>	<b>50</b>	<b>3,0</b>	<b>1,42857</b>	<b>47,619%</b>

**Análisis de la varianza:** Se realiza un análisis de varianza y se obtiene que La razón-F, que en este caso es igual a 4,91, es el cociente entre el estimado entre-grupos y el estimado dentro-de-grupos. Puesto que el valor-P de la prueba-F es menor que 0,05, existe una diferencia estadísticamente significativa entre las medias de los 5 niveles del factor con un nivel del 95,0% de confianza.

**Tabla 7. Resumen de varianza ANOVA**

<b>Fuente</b>	<b>Suma de Cuadrados</b>	<b>Gl</b>	<b>Cuadrado Medio</b>	<b>Razón-F</b>	<b>Valor-P</b>
Entre grupos	30,4	4	7,6	4,91	0,0023
Intra grupos	69,6	45	1,54667		
Total (Corr.)	100,0	49			

### **Análisis de las medias:**

Con el fin de identificar cuáles son los escenarios diferentes se utiliza la prueba Tukey HSD. Al analizar la gráfica se muestra que hay diferencia entre las medias de los escenarios:

Esc 1 - Esc 3 - Esc4;

Esc 2 - Esc 4;

Esc 3 - Esc 1;

Esc4 - Esc1 - Esc 2.

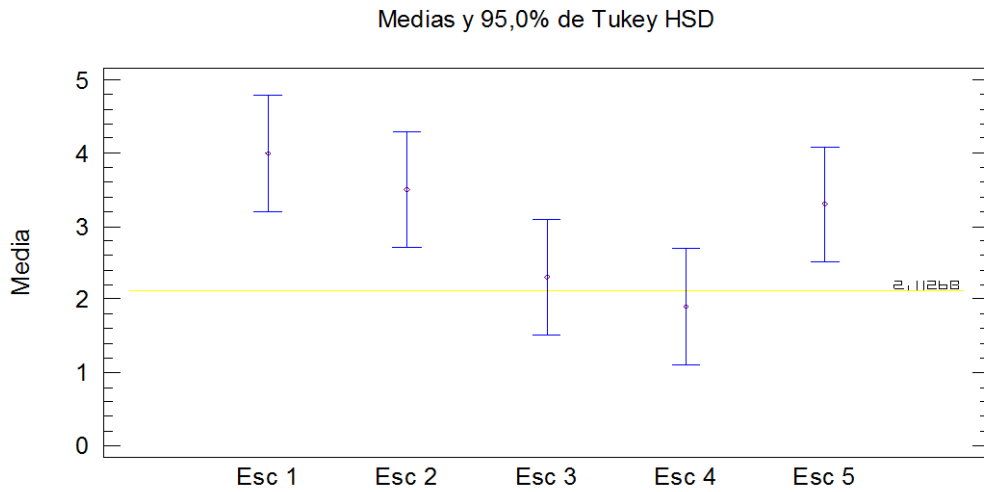


Figura 18. Grafica estadística de Medias de clasificación Vs Escenarios de Tukey HSD

Se puede observar 3 grupos homogéneos de escenarios según la prueba de TUKEY: el primer grupo corresponde a : Esc 4, Esc 3, Esc 5; el segundo grupo corresponde las medias de los escenarios Esc 3, Esc 5, Esc 2; y el tercer grupo corresponde a los Esc 5, Esc2 y Esc1. Por lo anterior las medias que no son homogéneas son Esc 4 y Esc 1.

Tabla 8. Prueba de homogeneidad

Escenarios	Casos	Media	Grupos Homogéneos
Escenario 4	10	1,9	X
Escenario 3	10	2,3	XX
Escenario 5	10	3,3	XXX
Escenario 2	10	3,5	XX
Escenario 1	10	4,0	X

## Análisis de los residuos:

Tabla 9. Análisis de los residuos

Prueba	Estadístico	Valor-P
Estadístico W de Shapiro-Wilk	0,970427	0,38944

Es importante comprobar el supuesto de normalidad. Debido a que el valor-P más pequeño de las pruebas realizadas es mayor ó igual a 0,05, no se puede rechazar la idea de que RESIDUOS se comportan o siguen una distribución normal con 95% de confianza. La grafica de normalidad la siguiente tendencia:

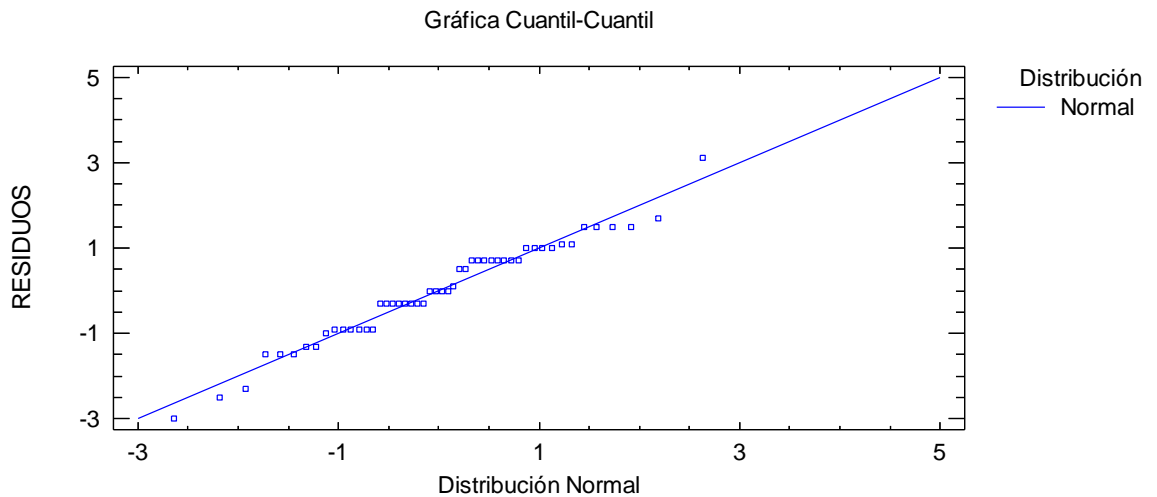


Figura 19. Grafica distribución normal para residuos

## Varianza Constante

Para comprobar este supuesto se debe realizar una prueba de Barlett, en la que se definen las siguientes hipótesis:

$$H_0: \sigma_1^2 = \sigma_2^2 = \dots = \sigma_k^2 = \sigma^2$$
$$H_0: \sigma_i^2 \neq \sigma_j^2 \text{ para algún } i \neq j$$

Para verificar que las varianzas son constantes se debe graficar los valores predichos contra los residuos. Si los puntos se muestran de manera aleatoria sin ningún patrón claro y contundente no se acepta el supuesto de que los tratamientos tienen igual varianza.

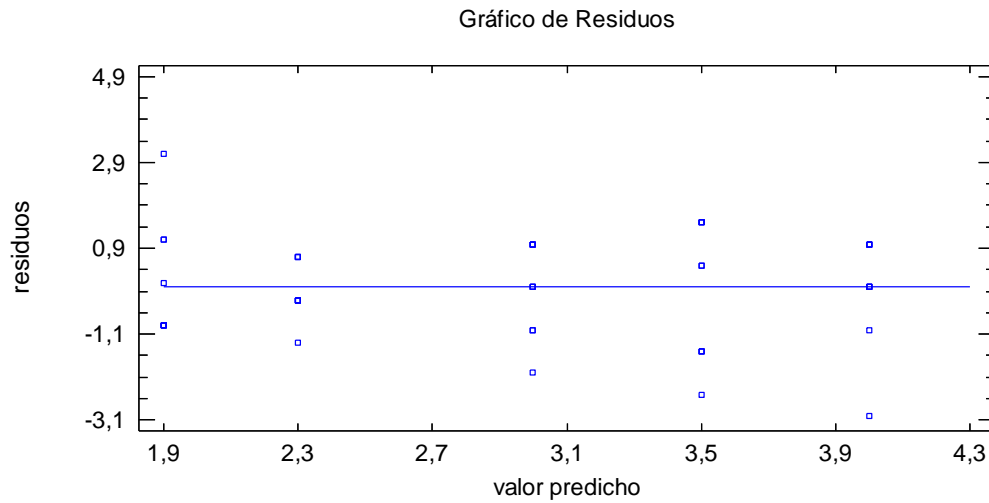


Tabla 10. Verificación de varianza

	<i>Prueba</i>	<i>Valor-P</i>
Prueba de Bartlett	1,14627	0,208168

Puesto que el valor-P es mayor o igual que 0,05, no existe una diferencia estadísticamente significativa entre las desviaciones estándar, con un nivel del 95,0% de confianza.

### Independencia

La suposición de independencia en los residuos puede verificarse si se grafica el orden en que se colectó un dato contra el residuos correspondiente. Como muestra la gráfica el comportamiento de los puntos es aleatorio, por lo que no hay razón para sospechar cualquier violación del supuesto de independencia.

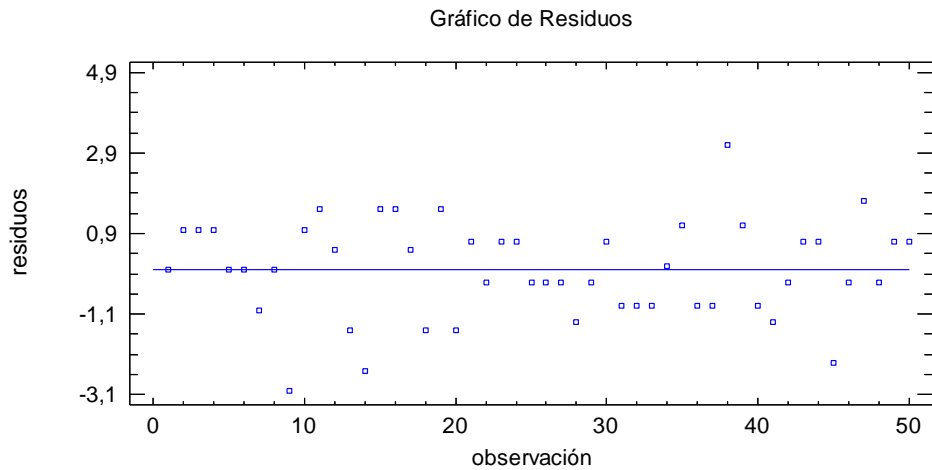


Figura 21. Grafica de independencia clasificación

#### 4.7.2 Definición del problema y factores de Valoración.

Se estudia la valoración de los 10 expertos sobre los 5 escenarios propuestos de la probabilidad de ocurrencia de los escenarios. A continuación se detalla la tabla de valoración obtenida:

Tabla 11. Respuesta de valoración por los expertos en los escenarios

Expertos	Escenarios				
	1	2	3	4	5
1	2	1	5	1	5
2	1	3	5	5	2
3	1	5	5	5	2
4	1	5	4	5	3
5	5	2	5	5	5
6	4	1	5	4	3
7	2	2	3	4	1
8	4	5	5	4	4
9	4	1	3	4	3
10	1	5	5	4	4

**Factor de Interés:** probabilidad de ocurrencia de los escenarios

**Niveles del factor:** 5 escenarios (Escenario A, Escenario B, Escenario C, Escenario D, Escenario E)  $K = 5$ .



**VARIABLES DE INTERÉS:** = Valoración de los Expertos sobre los métodos de estimación.

**Numero de réplicas (Expertos):** n = 10

**Numero de observaciones:** 50

### Hipótesis del problema

$\mu_1$  = Media de la valoración de los expertos en el escenario A.

$\mu_2$  = Media de la valoración de los expertos en el escenario B.

$\mu_3$  = Media de la valoración de los expertos en el escenario C.

$\mu_4$  = Media de la valoración de los expertos en el escenario D.

$\mu_5$  = Media de la valoración de los expertos en el escenario E.

Dónde:

$H_0$ :  $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = 0$  ; Se rechaza  $H_0$  en función de  $H_1$

$H_1$ :  $\mu_i \neq \mu_j$  para algunos  $i, j$

La hipótesis nula se define como la igualdad de las medias de la valoración de cada uno de los expertos en los diferentes escenarios y la hipótesis alternativa se define como la diferencia de al menos dos de las medias de valoración de los expertos en los diferentes escenarios.

En caso de no rechazar  $H_0$  se concluye que las medias de valoración de los 5 escenarios son estadísticamente iguales; pero si se rechaza, se concluye que al menos dos de ellos son diferentes.

### Modelo Estadístico:

$$Y_i = \mu + \tau_i + \varepsilon_i$$

### Análisis de los resultados obtenidos:

A continuación se muestran las valoraciones obtenidas para cada uno de los escenarios, se puede observar que el Escenario 3 es el que más consenso genera en el panel experto ya que su desviación estándar es una de las menores y su promedio en la valoración es de 4,5 (de 5). Igualmente el escenario 4 obtuvo una valoración importante con una desviación estándar de 1,19. Es así que en el análisis de clasificación y valoración los escenarios que obtuvieron calificaciones significativas son el 3 y el 4.

Tabla 12. Resumen estadístico para el factor de valoración

	Recuento	Promedio	Desviación Estándar	Coefficiente de Variación
Escenario 1	10	2,5	1,58114	63,2456%
Escenario 2	10	3,0	1,82574	60,8581%
Escenario 3	10	4,5	0,849837	18,8853%
Escenario 4	10	4,1	1,19722	29,2005%
Escenario 5	10	3,2	1,31656	41,1425%
<b>Total</b>	<b>50</b>	<b>3,46</b>	<b>1,5281</b>	<b>44,1649%</b>

### Análisis de la varianza:

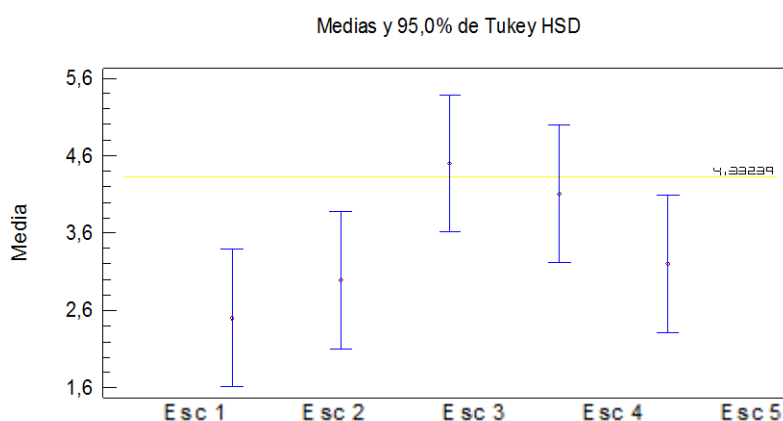
La razón-F, que en este caso es igual a 3,46 es el cociente entre el estimado entre-grupos y el estimado dentro-de-grupos. Puesto que el valor-P de la prueba-F es menor que 0,05, existe una diferencia estadísticamente significativa entre las medias de los 5 niveles del factor con un nivel del 95,0% de confianza.

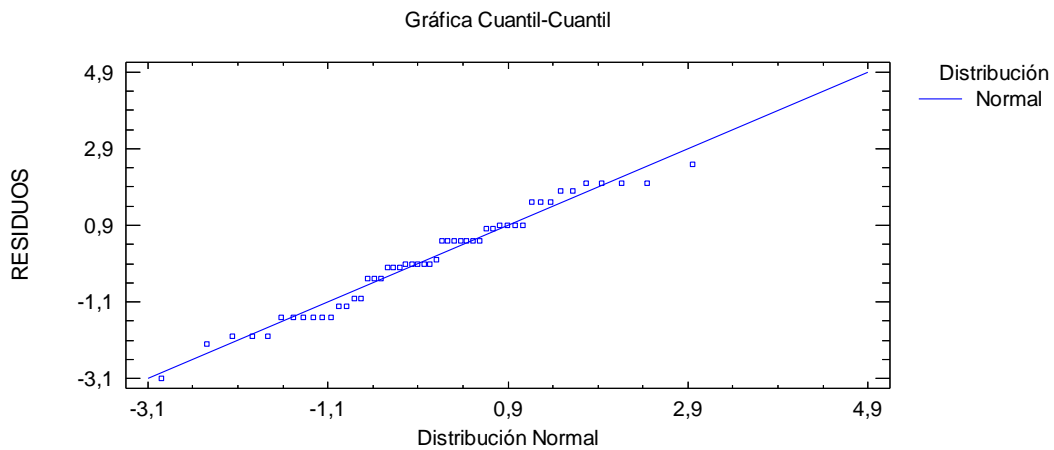
Tabla 13. Análisis de varianza ANOVA

Fuente	Suma de Cuadrados	GI	Cuadrado Medio	Razón-F	Valor-P
Entre grupos	26,92	4	6,73	3,46	0,0151
Intra grupos	87,5	45	1,94444		
Total (Corr.)	114,42	49			

### Análisis de Medias

Al analizar la gráfica de Medias se puede observar los escenarios que hay una diferencia estadística son: Esc 1 – Esc 3.





**Tabla 14. Análisis de Homogeneidad**

	<b>Casos</b>	<b>Media</b>	<b>Grupos Homogéneos</b>
Escenario 1	10	2,5	X
Escenario 2	10	3,0	XX
Escenario 5	10	3,2	XX
Escenario 4	10	4,1	XX
Escenario 3	10	4,5	X

En la tabla de prueba de medias se puede evidenciar que no existen diferencias estadísticamente significativas entre aquellos niveles que compartan una misma columna de X's, por lo anterior se muestra que los únicos escenarios que no se traslapan son el 1 y el 3.

**Análisis de residuos:**

El ANOVA analiza los supuestos de igualdad de varianza, normalidad e independencia. Para comprobar el supuesto de normalidad se aplica la prueba de Shapiro-Wilk, donde al ser el valor-P mayor o igual a 0,05, no se puede rechazar la idea de que RESIDUOS se comportan o siguen una distribución normal con 95% de confianza. Lo anterior se sustenta al graficar los residuos vs la distribución normal.

**Tabla 15. Análisis de residuos**

<b>Prueba</b>	<b>Estadístico</b>	<b>Valor-P</b>
Estadístico W de Shapiro-Wilk	0,963488	0,216204

**Figura 24. Gráfica distribución normal para residuos**

### Igualdad de varianza:

Para comprobar este supuesto se debe realizar una prueba de Barlett, en la que se definen las siguientes hipótesis:

$$H_0: \sigma_1^2 = \sigma_2^2 = \dots = \sigma_k^2 = \sigma^2$$

$$H_0: \sigma_i^2 \neq \sigma_j^2 \text{ para algún } i \neq j$$

Para verificar que las varianzas son constantes se debe graficar los valores predichos contra los residuos. Si los puntos se muestran de manera aleatoria sin ningún patrón claro y contundente no se acepta el supuesto de que los tratamientos tienen igual varianza.

Tabla 16. Prueba de varianza

Análisis de Varianza	Prueba	Valor-P
de Bartlett	1,13201	0,253927

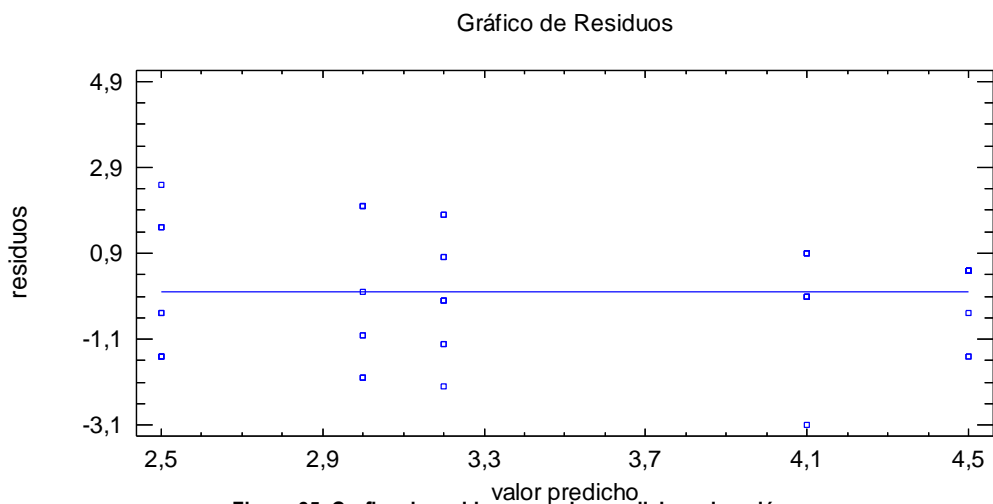


Figura 25. Grafica de residuos vs valor predicho valoración

### Independencia

La suposición de independencia en los residuos puede verificarse si se grafica el orden en que se colectó un dato contra el residuos correspondiente. Como muestra la gráfica el comportamiento de los puntos es aleatorio, por lo que no hay razón para sospechar cualquier violación del supuesto de independencia.

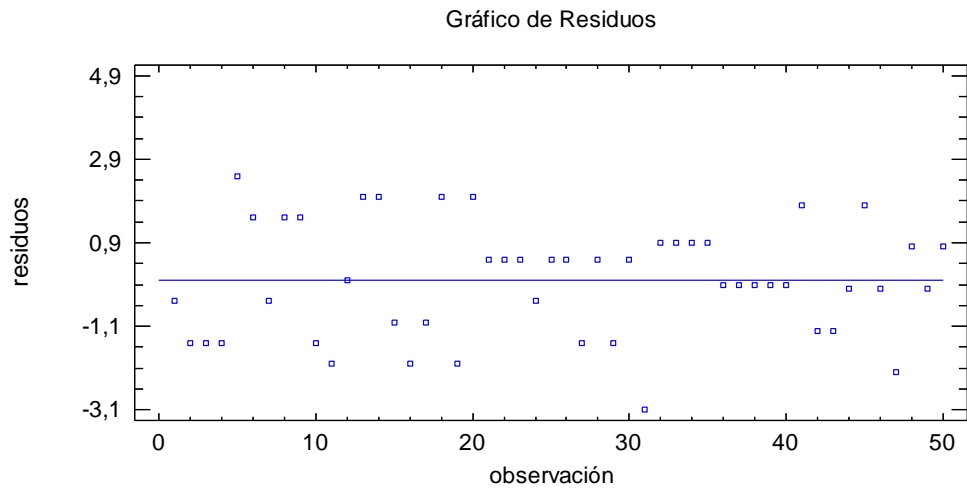


Figura 26. Grafica de independencia clasificación

#### 4.7.3 Resumen de los resultados obtenidos

El análisis estadístico evidencia el consenso de los expertos en los escenarios 3 y 4 tanto en las escalas de clasificación y valoración. Se destaca las observaciones realizadas, específicamente por el experto 8 al el cual asegura que con la base de datos historia se logra un mejor estimado, pero se debe tener en cuenta el programador que se usaría para estos estimados, esto en término de su experiencia con el lenguaje de programación y familiaridad del proyecto. El experto No 2 comenta que basan sus estimaciones en técnicas de descomposición teniendo en cuenta el número de requerimientos, tamaño de las tareas en cada requerimiento, número y complejidad de las interfaces y la prioridad por tareas. Las anteriores observaciones serán tenidas en cuenta para el diseño del modelo de mejora de procesos de estimación de costos de software.

## CAPITULO V

### 5. DISEÑO DE MODELO DE MEJORA DE PROCESOS DE ESTIMACIÓN DE COSTOS DE SOFTWARE PARA LAS EMPRESAS DESARROLLADORAS

#### 5.1 Introducción

En este capítulo se presenta el modelo para mejorar los procesos de estimación de costos de Software de las empresas desarrolladoras, partiendo de la base conceptual y del estado del arte de los métodos más utilizados de estimación de costos de software (Da Yang et al., 2008). El modelo planteado para mejorar los procesos de estimación se diseña teniendo en cuenta los dos escenarios que obtuvieron mejor calificación y valoración, donde se realizó un análisis experimental que se describe en el capítulo 4.

#### 5.2 Descripción de los Componentes del modelo

Se establecieron 3 agentes para el modelo, que se definen a continuación detalladamente. En este modelo se definen 3 categorías las cuales representan el estado de cumplimiento de los procesos para lograr estimaciones mas precisas: Alto, medio, bajo. A continuación se detalla las escalas propuestas:

##### 1. METODOLOGÍA

La metodología se refiere al marco de trabajo que utilizan las empresas para planificar y estructurar el proceso de desarrollo de software. Existen diferentes modelos propuestos para desarrollar un producto de software de calidad; cada modelo define un ciclo de vida cuyo objetivo es generar una serie de pasos aplicados para lograr un producto de software. La metodología que utiliza este modelo tiene un enfoque interactivo incremental, donde el cliente obtiene beneficios en forma incremental y se realizan entregas parciales por medio de los llamados Sprint.

**Categoría Baja:** Se establecen los requisitos y alcance del proyecto con el cliente, pero no existe claridad por problemas en la comunicación. Los requisitos se descomponen en actividades o tareas más específicas pero se muestra retraso para finalizar la iteración al ser poco clara o presentar inconvenientes técnicos. En este nivel los requisitos tienen a ser volátiles atrasando la evolución del producto.

Un porcentaje significativo del personal no cuenta con experiencia en proyectos similares y no existe una base de datos histórica que sirva como referencia en los procesos de estimaciones de tiempo y complejidad de la tarea. A pesar de que la empresa maneja una documentación para el cliente, es poco clara y no se construye al terminar cada sprint sino al finalizar todo el proyecto.

**Categoría Media:** Se realiza la toma de requisitos y alcance con el cliente. No existe claridad en lo que desea al presentarse problemas en la comunicación, por lo que el equipo establece unos requerimientos iniciales (Producto Mínimo Viable). Se descomponen los requerimientos por el nivel de complejidad donde el equipo de trabajo cuenta con experiencia empírica en proyectos similares, pero no posee una base de datos formal de todos los detalles de esos proyectos. A pesar que la empresa maneja documentación poco clara y no se realiza al terminar cada sprint sino al finalizar todo el proyecto.

**Categoría Alta:** Se gestiona las expectativas del cliente y futuros usuarios mediante reuniones que definen el alcance y se establecen los requisitos iniciales del proyecto de software. Se muestra una clara descomposición de requisitos en actividades o tareas más específicas, donde el cliente identifica prioridades a los requerimientos de alto impacto con el fin de que se generen entregables en el menor tiempo posible. Se maneja una base de datos histórica para generar estimados más acorde a la realidad que almacena información de proyectos anteriores (estimados, tiempo por tarea, número de personas por mes, etc). En cada sprint va creciendo el manual del usuario para que el cliente pueda conocer de manera detallada como funciona producto generado.

## 2. CAPACIDAD DE EQUIPO DE TRABAJO

La capacidad del equipo de trabajo es el insumo mas importante ya que se refiere a la relación del número de personas trabajando para un proyecto y partiendo del rol que desempeña cada uno para su ejecución, las habilidades y experiencia adquirida el esfuerzo va a ser mucho menor para la ejecución del proyecto.

**Categoría Baja:** No se muestra una organización por roles definidas del equipo de trabajo, generando inestabilidad y problemas para asumir actividades. No existe una política de capacitación al personal, pero el personal ha adquirido los conocimientos de forma empírica. La comunicación con el cliente no está basada en el principio de la colaboración lo que genera poca retroalimentación a inquietudes al equipo por parte del cliente.

El líder del proyecto no cumple con el cronograma de reuniones con su equipo de trabajo lo que genera tareas pendientes para futuras interacciones y el desconocimiento del estado actual del ciclo de vida de desarrollo.

**Categoría Media:** Se cuenta con un equipo base pero no están bien definidos los roles. No existe una política clara de capacitación de personal aunque existe el interés de crearla. Se

maneja un seguimiento poco robusto de actividades de los desarrolladores. Se presentan problemas de comunicación en el equipo de trabajo para responder inquietudes de funcionalidad por el tiempo de respuesta o barrera tecnológica. Se realizan tareas de seguimiento con el equipo de trabajo.

**Categoría Alta:** La empresa cuenta con un equipo con roles bien definidos, el cual se encuentra altamente capacitado sobre las diferentes herramientas que se utilizan en cada uno de los proyectos. Existen políticas de capacitación para elevar el nivel de competitividad de los trabajadores. El cliente es parte activa del proceso de desarrollo, el cual puede tomar decisiones en cada sprint. Las reuniones del líder del proyecto con el equipo de trabajo son altamente efectivas al evaluar el cronograma de entrega, el cumplimiento de los estimados y la solución a bloqueos.

### 3. PRODUCTOS TECNOLÓGICOS

Los productos tecnológicos hacen referencia a las herramientas que apoyan el proceso de desarrollo del software. Es importante resaltar que existen herramientas de software propietario o con licencia y herramientas libres, cuya preferencia en ocasiones es determinada por el cliente o si la empresa desarrolladora posee un producto que comercializa o con la cual tiene experiencia puede decidir qué tipo de herramienta utilizar.

**Categoría Baja:** Utilizar tecnologías para la arquitectura del sistema, patrones de desarrollo con abstracción de capas, lenguajes de programación obsoletos o que no son una solución apropiada para el proyecto.

Además de no tener diferentes environments de desarrollo, control de versión, Branching o manejarlos de forma deficiente.

**Categoría Media:** Utilizar tecnologías para la arquitectura del sistema, patrones de desarrollo con abstracción de capas, lenguajes de programación por ser lo que siempre se ha usado y es en lo que se tiene experiencia aunque otras tecnologías pueden ser una mejor solución a todos los parámetros anteriores. Además de no tener diferentes environments de desarrollo, control de versión, Branching o manejarlos de forma deficiente.

**Categoría Alta:** Utilizar tecnologías para la arquitectura del sistema, patrones de desarrollo con abstracción de capas, lenguajes de programación en el estado del arte y que son una selección apropiada para el proyecto. Tener environments de desarrollo, control de versión, Branching y manejarlos de forma eficiente.



## Categorías del agente metodología

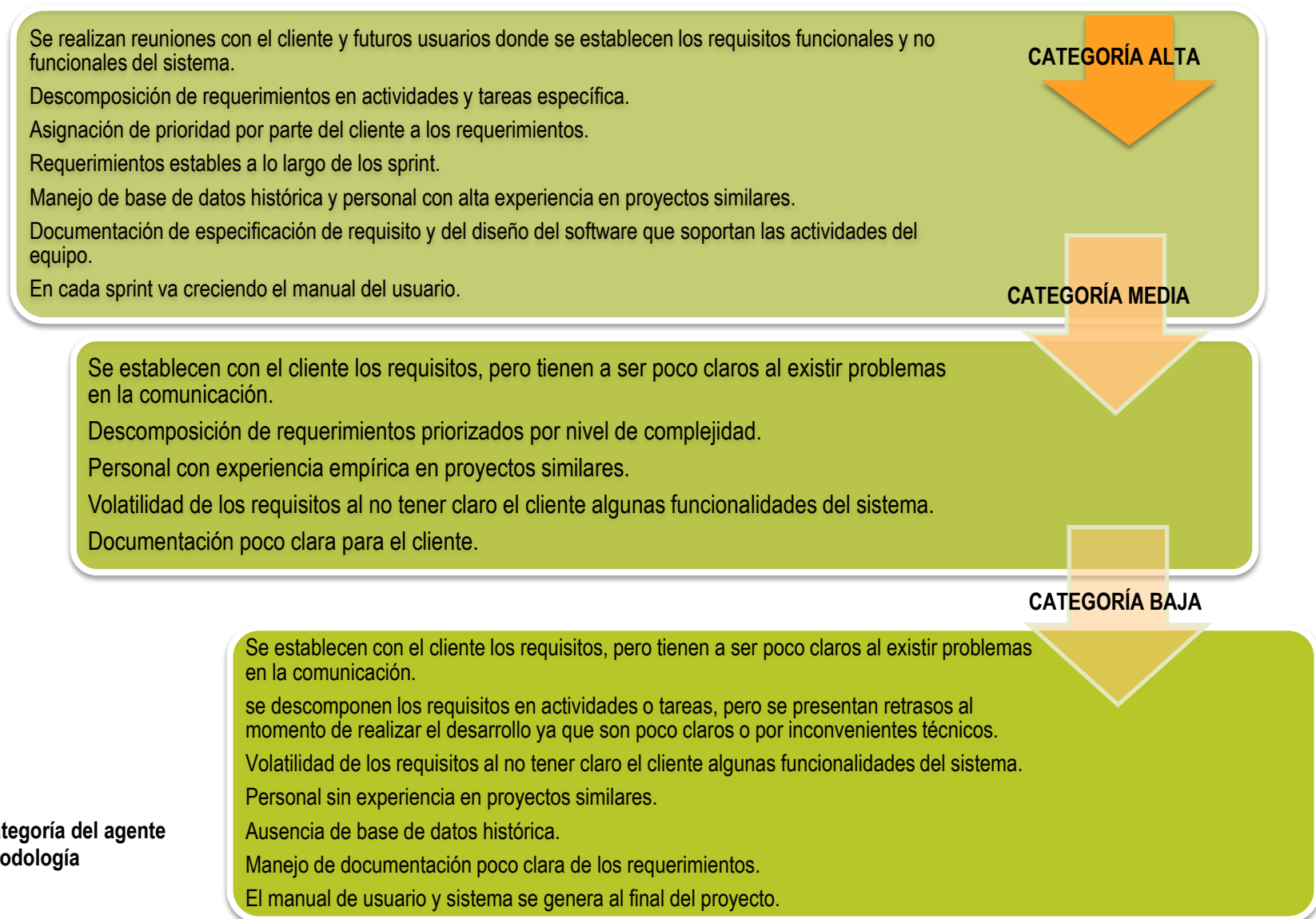


Figura 27 Categoría del agente Metodología

## Categorías del agente Equipo de trabajo

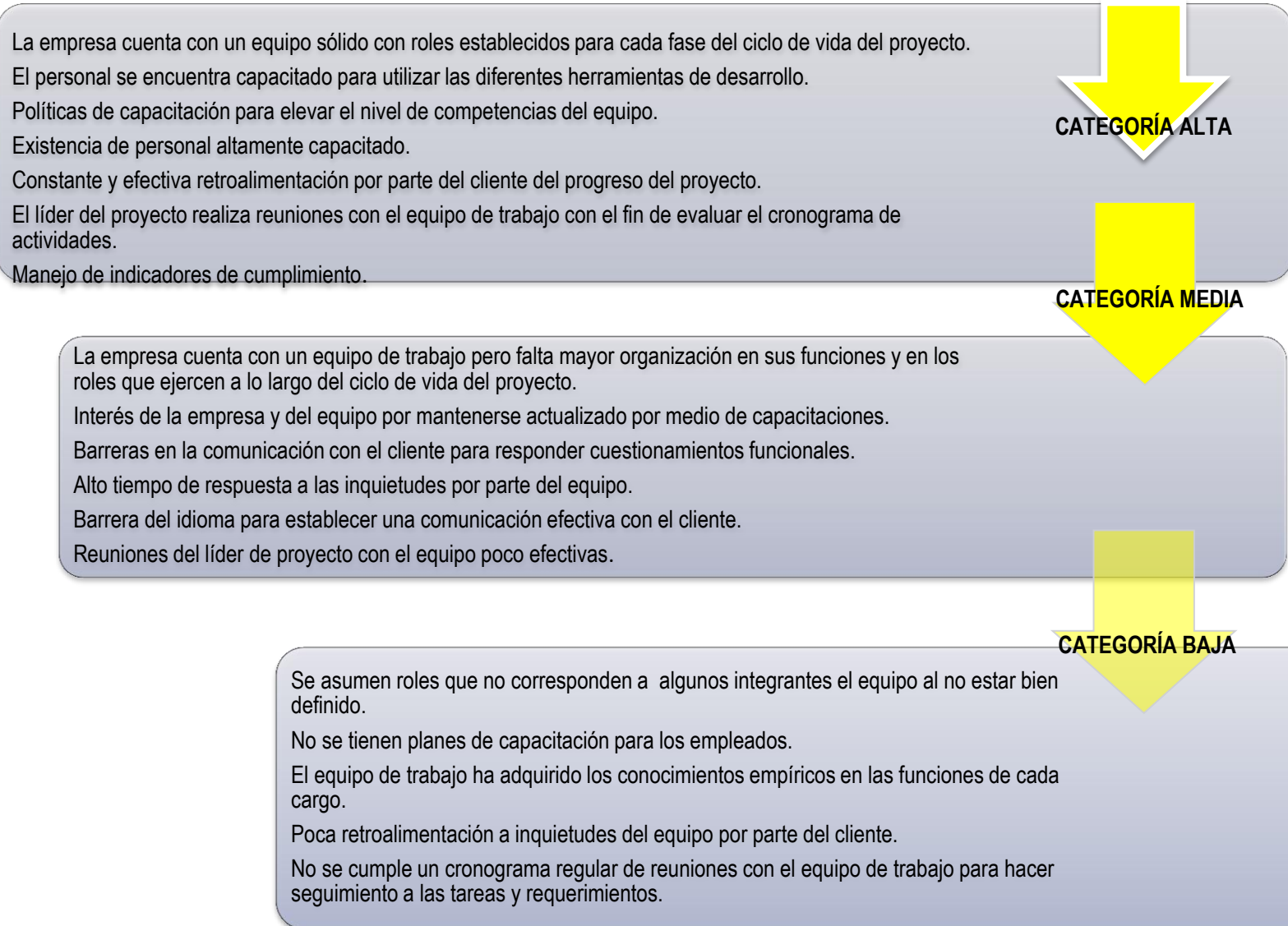


Figura 28. Categoría del agente equipo de trabajo

## Categoría del agente Herramientas tecnológicas

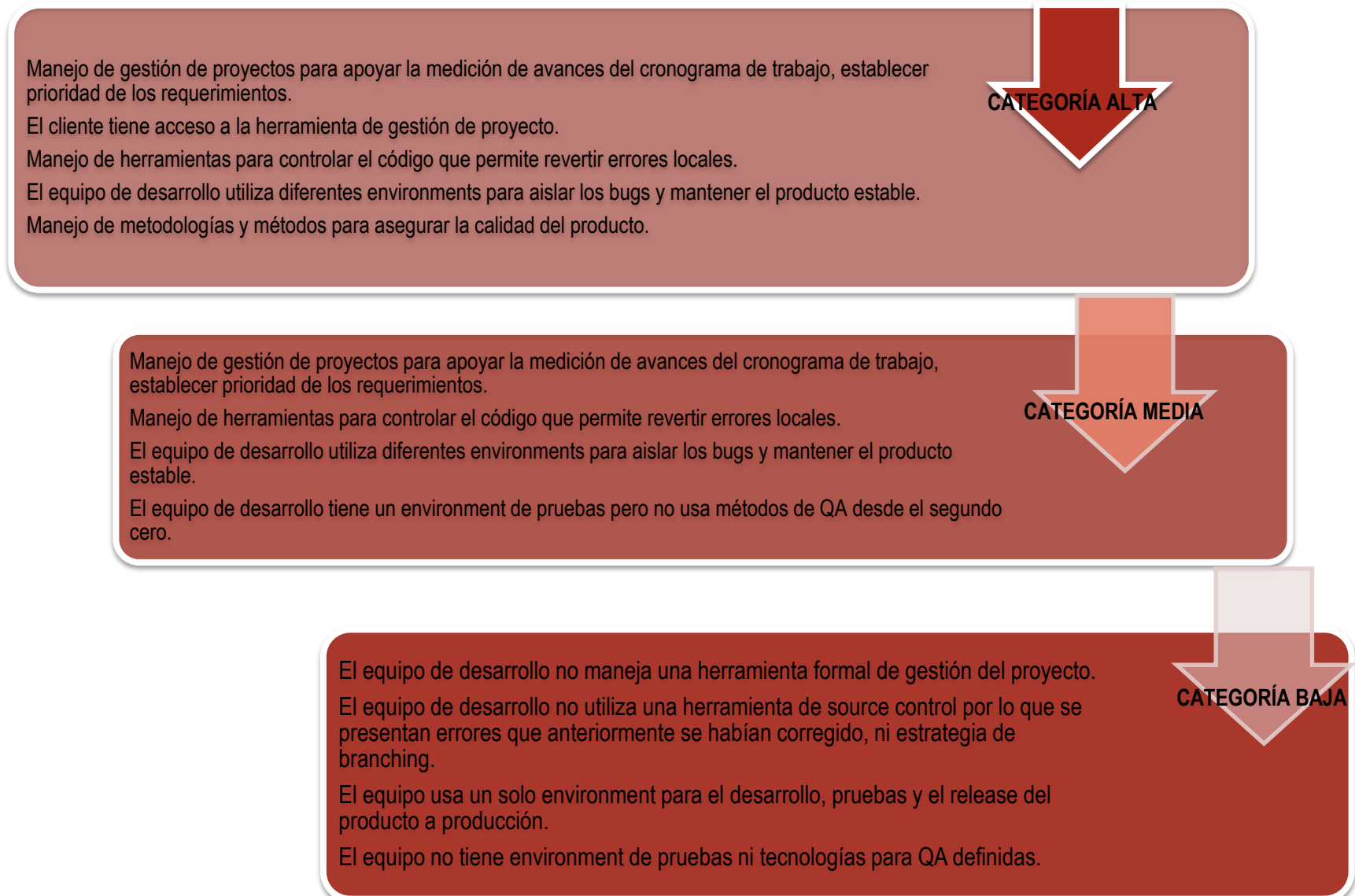


Figura 29. Categoría del agente Herramientas Tecnológicas

### 5.3 Ponderación de Categorías

Todos los factores son igualmente importantes, por los que el factor de metodología, capacidad de equipo de trabajo y herramientas tecnológicas tienen un porcentaje del 100% cada uno dentro de la estructura del modelo propuesto. A continuación se detalla cada una de las variables que alimentan a los agentes. De acuerdo a la retroalimentación con los expertos de cómo se realizan las estimaciones de costos de un proyecto de software, se identifican por cada uno de los agentes del modelo una lista de variables críticas en el desarrollo del software, donde cada una afecta de la misma manera la precisión de las estimaciones de un proyecto, por lo cual el porcentaje total se divide entre el número de estas variables. Igualmente al analizar los factores de clasificación y valoración de los escenarios evaluados por los expertos, se encontró evidencia estadística de que al menos un par de medias son iguales, en este caso los escenarios 3 y 4 que son la base para el modelo propuesto, donde las características y variables son consideradas con igualdad de peso.

Tabla 17. Variables del agente Metodología

Variables del agente Metodología				
Alcance y requerimientos	Descomposición	Estabilidad de los requerimientos	Histórico de proyectos	Documentación
Establecimiento de requerimientos del Sistema.	Se descomponen los requisitos en tareas o funcionalidades más pequeñas	Requerimientos estables a lo largo del sprint.	Manejo de Base de datos histórica y personal con alto conocimiento en proyectos similares.	Se maneja una documentación y manuales sobre el proyecto que se realiza
<b>20%</b>	<b>20%</b>	<b>20%</b>	<b>20%</b>	<b>20%</b>
<b>Porcentaje total: 100%</b>				

A continuación se presenta la ponderación para cada una de las variables del agente Metodología:

Variable	Escala de la variable	Puntos
Alcance y requerimiento.	Se establecen los requisitos funcionales y no funcionales del sistema de manera clara y específica a partir de reuniones con el cliente y futuros usuarios. Cada requisito se debe completar en una iteración.	20
	Se establecen los requisitos funcionales y no funcionales del sistema a partir de reuniones con el cliente, pero por problemas de comunicación el requerimiento no se traduce a lo que el usuario final requería.	10
	Se establecen los requisitos funcionales y no funcionales del sistema de manera poco clara y específica ya que el cliente no está seguro de cómo debe funcionar el sistema de información.	5

Descomposición	Al tener los requisitos claros, se descomponen en actividades o tareas y se le asigna un nivel de prioridad por parte del cliente con el fin generar entregables en el menor tiempo posible, que tengan alto valor para el cliente.	20
	Se descomponen los requisitos en actividades o tareas y se prioriza por el nivel de complejidad para su desarrollo.	15
	Se descomponen los requisitos en actividades o tareas, pero se muestran retraso al momento de realizar el desarrollo ya que son poco claros o por inconvenientes técnicos.	5
Estabilidad de los requerimientos	No existen cambios significativos de los requerimientos a lo largo del ciclo de desarrollo.	20
	Se presentan volatilidad de los requisitos al no tener claro el cliente las funcionalidades del Sistema.	10
Histórico de proyectos	El equipo de desarrollo cuenta con una base de datos histórico donde se almacena información relacionada con la descripción del requerimiento, complejidad del requerimiento, tiempo estimado de una actividad, tiempo real de la actividad, duración general del proyecto en horas, valor por hora dependiendo del perfil del desarrollador, etc, así como personal con alta experiencia en proyectos similares.	20
	El equipo de desarrollo cuenta con experiencia empírica en proyectos similares, pero no posee una base de datos formal de todos los detalles de esos proyectos.	15
	El equipo de desarrollo no cuenta con una base de datos histórica para almacenar información de los proyectos o experiencia en proyectos similares por lo tanto su capacidad de estimar genera alta incertidumbre.	0
Documentación	Para la empresa desarrolladora es primordial documentar la especificación de requisitos (SRD) que contiene toda la especificación del sistema, así como el diseño del software en herramientas disponibles para todo el equipo. En cada sprint debe ir creciendo el manual de usuario.	20
	No se documenta con claridad para el cliente el manual del sistema y del usuario.	10
	No se tiene en cuenta llevar la documentación clara de los requerimientos y el manual de usuario se genere al final del proyecto.	5

Tabla 18. Variables del agente equipo de trabajo

Variables del agente equipo de trabajo			
Organización del equipo	Competencias del equipo del trabajo	Comunicación con el cliente	Seguimiento.
Equipo de trabajo organizado en roles para cumplir las actividades de desarrollo.	Personal capacitado y existen políticas definidas de capacitación de personal.	Fluida y continua comunicación con el cliente en las actividades de desarrollo.	Reuniones efectivas de seguimiento interno.
<b>25%</b>	<b>25%</b>	<b>25%</b>	<b>25%</b>
<b>Porcentaje total: 100%</b>			

A continuación se presenta la ponderación para cada una de las variables del agente equipo de trabajo:

Variable	Escala de la variable	Puntos
Organización del equipo	La empresa cuenta con un equipo base para cumplir las labores de la implementación de la metodología seleccionada. Los diferentes roles dependerán tanto de la metodología como de las tecnologías que impliquen la arquitectura. También se pueden combinar más de un rol en una sola persona.	25
	La empresa cuenta con un equipo de desarrollo, pero no están definidos todos los roles necesarios para llevar a cabo las labores de implementación de la metodología seleccionada.	12,5
	Los miembros del equipo de desarrollo tienen que cumplir labores que no les corresponden inicialmente al no tener un equipo definido, y/o no se respetan los roles definidos.	0
Competencias del equipo de trabajo	Existencia de personal capacitado para utilizar las diversas herramientas que van a ayudar al equipo a trabajar conforme a la metodología seleccionada. Programas de capacitación para elevar el nivel de competencia.	25
	Se muestra el interés de la empresa y del equipo por ser capacitados para cumplir con los requerimientos tecnológicos del proyecto.	15
	Personal con poca competencia sobre los requerimientos tecnológicos del proyecto y la empresa no cuenta con una política de capacitación de personal.	0
Comunicación con el cliente	Constante retroalimentación por parte del cliente del progreso del proyecto.	25
	La comunicación con el cliente se hace difícil sea porque este no posee los conocimientos de primera mano para responder a los cuestionamientos funcionales por parte del equipo de trabajo, por la barrera del idioma o alto tiempo de respuesta a las inquietudes.	12,5

	Poca o insuficiente retroalimentación a inquietudes del equipo por parte del cliente.	3,12
Seguimiento	El líder del proyecto realiza reuniones diarias con el equipo para evaluar el cronograma de trabajo. Se evalúan cumplimiento de estimados y solución de bloqueos.	25
	Reuniones del líder del proyecto con el equipo de trabajo que son poco efectivas en culminar las tareas dentro de su estimación.	12,5
	No se cumple un programa regular de reuniones con el equipo de trabajo para hacer seguimiento a las tareas y requerimientos.	0

Tabla 19. Variables del agente herramientas tecnológicas

Variables del agente Herramientas Tecnológicas			
Apoyo a la gestión	Source control y Branching	Enviroments	QA
Herramienta de apoyo a la gestión	Herramienta que permite revertir errores locales y generar trabajo simultaneo de varios desarrolladores en una misma actividad o tarea.	Manejo de diferentes ambientes de desarrollo para aislar los errores y mantener el producto establecer.	Manejo de métodos y metodologías de aseguramiento de la calidad.
25%	25%	25%	25%
<b>Porcentaje total: 100%</b>			

A continuación se presenta la ponderación para cada una de las variables agente herramientas tecnológicas:

Variable	Escala de la variable	Puntos
Apoyo a la gestión	El equipo de desarrollo utiliza herramientas de gestión de proyectos para apoyarse en la medición de los avances en el cronograma de trabajo, prioridad de los requerimiento, histórico del proyecto, fechas de entregables, etc. El cliente tiene acceso a esta información.	25
	El equipo de desarrollo no maneja una herramienta formal de gestión del proyecto.	0
Source control y Branching	El equipo de desarrollo utiliza herramientas de gestión de source control que permite revertir errores locales y el trabajo simultaneo de varios desarrolladores en una misma actividad o tarea. Igualmente poseen un estrategia para crear Branchs (nueva versión del proyecto) acorde a las necesidades del proyecto.	25

	El equipo de desarrollo no utiliza una herramienta de source control por lo que se presentan errores que anteriormente se habían corregido, ni estrategia de Branching.	0
Enviroments	El equipo de desarrollo utiliza diferentes environments (al menos Development, QA y producción) para mantener aislados los bugs y el producto estable.	25
	El equipo usa un solo environments para el desarrollo, pruebas y el release del producto a production.	12,5
QA	El equipo usa métodos y tecnologías de quality assurance desde el momento que se escribe código para una tarea como TTD (test driven development) y tiene al menos un environments de pruebas.	25
	El equipo de desarrollo tiene un environment de pruebas pero no usa métodos de QA desde el segundo cero.	12,5
	El equipo no tiene environments de pruebas ni tecnologías para QA definidas.	0

De acuerdo al modelo de mejora de procesos de estimación de costos de software, se plantea el siguiente esquema que representa claramente la realidad de los aspectos a tener en cuenta para realizar estimaciones mas cercana a la realidad. El siguiente grafico radial en cada una de sus partes tiene una escala para establecer la categoría del modelo.

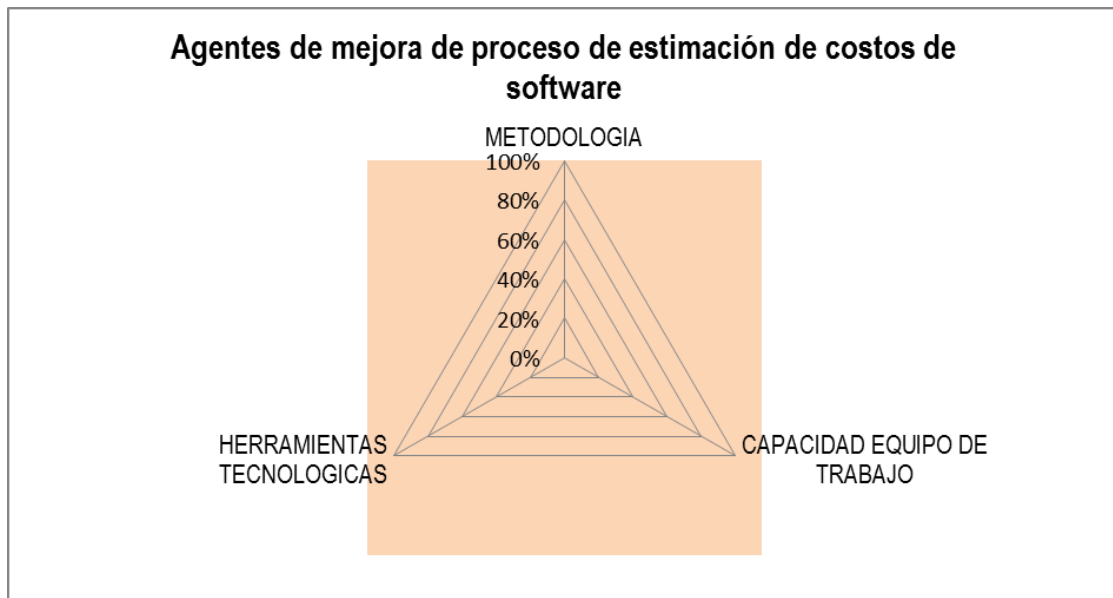


Figura 30. Representación los agentes de mejora del proceso de estimación de costos.

**Categorías del modelo:** A continuación se establece el rango de puntos para cada Categoría.



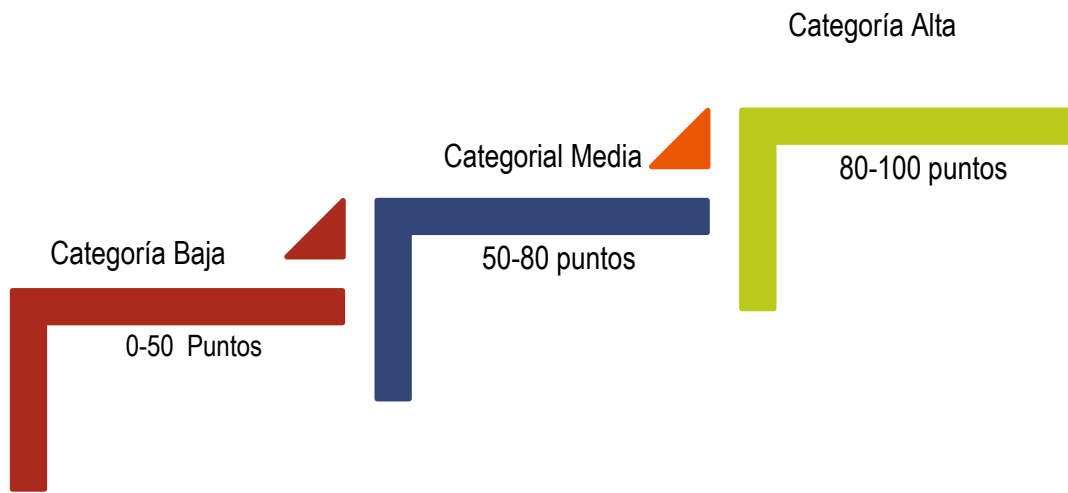
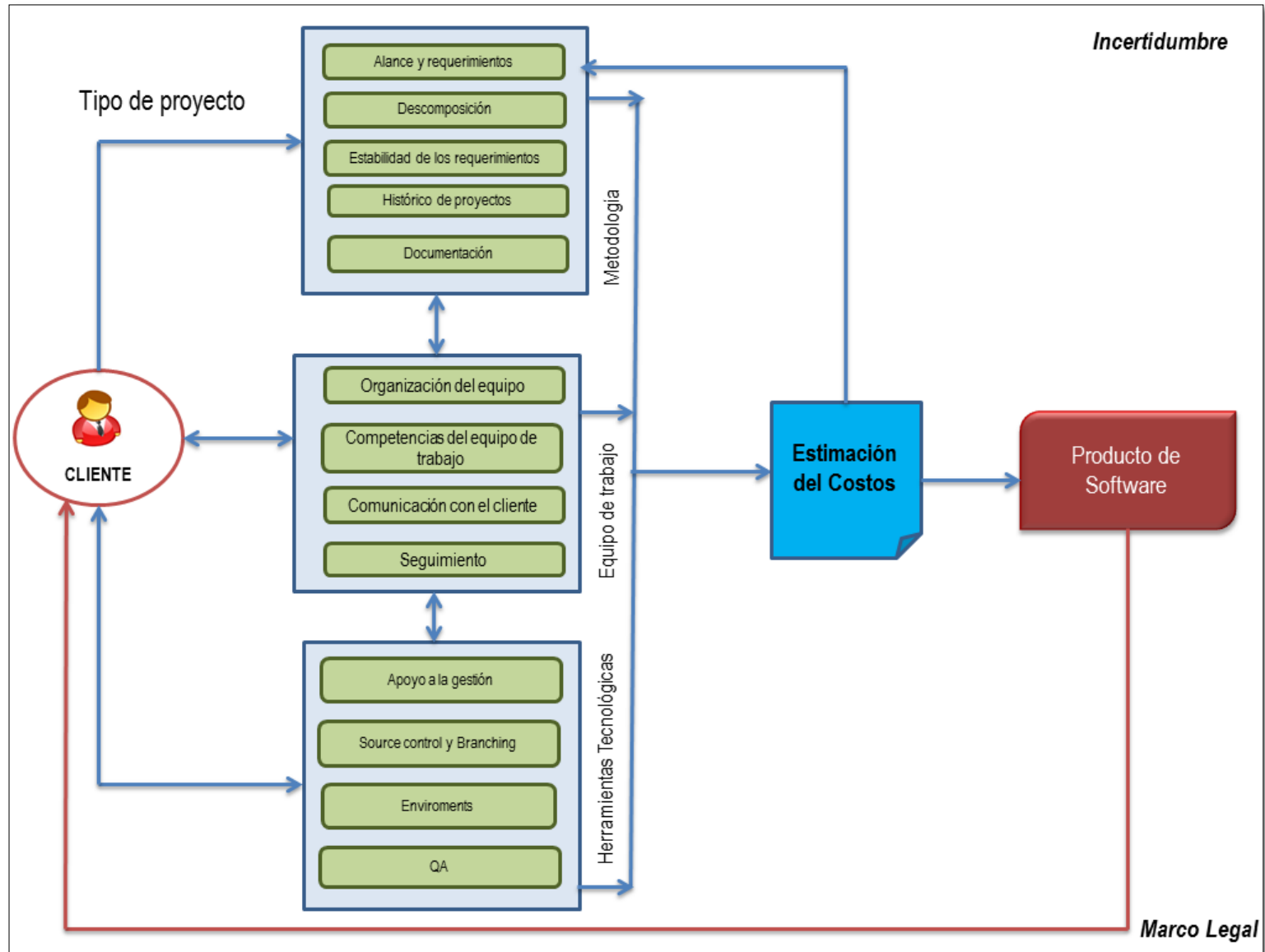


Figura 31. Categorías del modelo

## 5.4 Modelo conceptual de mejora de procesos de estimación de costos de software para las empresas desarrolladoras



## **Explicación del modelo conceptual:**

El modelo de mejora de procesos de estimación de costos de software se basa en 3 agentes fundamentales: Metodología, capacidad de equipo de trabajo y herramientas tecnológicas; donde el cliente es pieza principal del proceso de desarrollo al ser parte activa de este y tomar decisiones importantes a lo largo del ciclo de vida del proceso.

El proceso empieza cuando el cliente identifica que tipo de solución desea para su compañía, una solución a la medida o un software empaquetado. Cabe resaltar que un software empaquetado ya cuenta con algunas funciones necesarias para su implementación, siendo más complicada su adaptación al tener que ajustarlas a las necesidades del cliente.

El cliente adquiere un software empaquetado si cumple con los requerimientos que se ajustan a sus necesidades. Negocia el número de módulos que se van a implementar. Si los módulos no se ajustan completamente se realizan modificaciones según el requerimiento, siendo muy parecido el proceso al de un software a la medida. En ambos casos aplican los procesos de estimación de costos de software, por lo que es importante incluirlo en el modelo propuesto como "Tipo de proyecto" y se justifica su importancia en los dos escenarios valorados y clasificados por los panelistas.

La empresa desarrolladora de software en conjunto con el cliente define el marco de trabajo que se va a utilizar para implementar la solución. Se definen el alcance y los requerimientos. El cliente envía una lista de requerimientos específicos, pero en otras ocasiones el cliente no sabe exactamente lo que desea y va identificando estas necesidades a medida que se van entregando funcionalidades del sistema. Cabe aclarar que conocer los requisitos y el alcance son pieza clave para poder identificar el número de personas que van a participar en el proyecto. Para el autor (Sommerville, 2005) se debe tener presente que cualquier cambio en los requisitos del software significa inestabilidad en el costo y en la planificación temporal. En un enfoque iterativo es posible revisar las estimaciones a medida que se conoce más información del proceso y variarla cuando el cliente haga cambios.

Los requisitos se descomponen en actividades o tareas específicas y se le asigna un nivel de prioridad con ayuda del cliente para generar entregables en el menor tiempo posible. Esto es primordial para establecer el esfuerzo medido en el número de personas/mes que se necesitan en el proyecto.

Es importante para realizar las estimaciones contar con una base de datos histórica, por medio de analogías con proyectos similares se puede medir aspectos como complejidad de una tarea, duración de la tarea, tiempo estimado, etc, y por ello llegar a estimaciones de mayor exactitud.

Es importante la documentación en el agente metodología y se debe completar en cada interacción con el fin de que el cliente conozca a detalle la funcionalidad y guía de lo desarrollado, siendo este un aspecto para medir el nivel de satisfacción.

El cliente tiene relación directa con el agente "capacidad equipo de trabajo" al poder participar en la escogencia del número de recursos para el desarrollo de un proyecto. Si el equipo tiene los roles bien definido y se encuentran altamente capacitados en las herramientas de desarrollo y aquellas que soportan la gestión de su actividad el tiempo de respuesta va a ser más efectivo. Igualmente es

importante que existan políticas de capacitación y plan de incentivos al personal. Debe existir constante retroalimentación por parte del cliente del progreso del proyecto, lo que permitirá ajustar requisitos funcionales y no funcionales. El líder del proyecto debe conocer el progreso real del proyecto por medio de reuniones con su equipo donde se evalúe el cronograma de trabajo, los tiempos y la solución a bloqueos y a errores en la aplicación.

El agente de "herramientas tecnológicas" se basa en el uso de tecnologías para la arquitectura, patrones de desarrollo con abstracción de capas, lenguajes de programación apropiados para el proyecto. Igualmente se

tomar decisiones del desarrollo del proyecto. Por otra parte, al utilizar el equipo herramientas para controlar el código puede revertir errores locales, así como usar métodos para asegurar la calidad del producto.

Los factores anteriormente mencionados están relacionados y se debe tener en cuenta que cada una de las variables que hacen parte de estos, permite marcar la diferencia entre beneficios y pérdidas para la compañía desarrolladora al evaluar aspectos como tiempo, costo, recursos disponibles, etc.

El objetivo final es desarrollar un producto que satisfaga las necesidades del cliente en el tiempo deseado y a un valor acorde. Al basarse el modelo propuesto en un Sprint y sobre estos entregables el cliente evalúa lo obtenido y vuelve al inicio del proceso para refinar o agregar funcionalidades al sistema.

## **CAPITULO VI**

### **1. CONCLUSIONES Y TRABAJOS FUTUROS**

En este capítulo se establecieron las conclusiones del estudio y por último las sugerencias para futuras investigaciones.

#### **1.1 Conclusiones**

El modelo propuesto se basó en un conjunto de escenarios o posibles ambientes de como las empresas realizarían sus procesos de estimaciones de costos. Se seleccionaron expertos de diferentes empresas dedicadas a desarrollar productos de software para afinar las características de los escenarios o proponer según su experiencia una nueva forma de estimar los costos del software. Posterior a esto se realizó un proceso de clasificación y validación utilizando el método Delphi.

Al obtener el diagnóstico de la clasificación y valoración de cada escenario, se realizó un análisis experimental para soportar las respuestas cualitativas y obtener los dos escenarios con características similares lo que alimentó la creación del modelo de mejora de procesos de estimación de costos de software.

Las empresas participantes mostraron su interés de conocer el resultado del estudio, por lo que se les compartió el modelo con el fin de poder identificar su nivel e el factor de metodología, capacidad de equipo de trabajo y herramientas tecnológicas y cómo influyen para generar estimaciones de costos más precisas y poder establecer un valor de su producto al cliente.

Los métodos de estimación de costos consultados en la bibliografía en general son muy especializados y utilizan fórmulas matemáticas lo que tiende a ser muy abstracto y complicados para las empresas, alejándolas de su uso práctico. Sin embargo la precisión de las estimaciones es afectada en ocasiones por agentes subjetivos y las malas prácticas en la planificación del proyecto. Este modelo propone mejoras a tener en cuenta en los procesos de desarrollo para estimar los costos de un proyecto de manera mucho más realista al contemplar los componentes de metodología, Herramientas tecnológicas y equipo de trabajo.

Los escenarios planteados en la primera ronda corresponden a los métodos de estimación de costos de software más utilizados por las empresas desarrolladoras, tal y como lo aseguran los autores Da Yang et al. (2008) & Moløkken-Østvold et al. (2004) donde se destacan el juicio experto, estimación por analogía, precio para ganar y el uso de aplicaciones para soportar las estimaciones.

Cabe resaltar que el escenario 3 es una modificación del método del juicio experto ajustado a las necesidades del mercado. Según la literatura, este método no es el más recomendado al ser poco confiable, ya que las estimaciones no dependen de la funcionalidad del software, sino de lo que el cliente está dispuesto a pagar. De esta manera se evidencia que existe una falta de cultura de las empresas

desarrolladoras de no utilizar otros métodos para no perder opciones de clientes potenciales o mantener los consolidados.

En la primera ronda del método Delphi los expertos revisaron cada uno de los escenarios propuestos y tuvieron la opción de realizar comentarios y/o sugerencias; así como agregar y modificar características. Se evidencio que los métodos no se aplicaban en su totalidad, sino que se adaptaron a un contexto muy particular de acuerdo a las condiciones socio económico de la región caribe. Es por esto que para la segunda rondas, los diferentes métodos propuestos según la literatura consultada son modificados y refinados según la experiencia y aplicación en las empresas, siendo la base para la creación del modelo propuesto de mejora de procesos de estimación de costos de software.

En la primera ronda del método Delphi los expertos revisaron cada uno de los escenarios propuestos y tuvieron la opción de adicionar comentarios y/o sugerencias; así como agregar y modificar características. Se evidencio que los métodos no se aplicaban en su totalidad, sino que se adaptaron a un contexto muy particular de acuerdo a las condiciones socio económico de la región caribe. Es por esto que para la segunda rondas, los diferentes métodos propuestos según la literatura consultada son modificados y refinados según la experiencia y aplicación en las empresas.

Es importante aclarar que se trata de una investigación cualitativa y por lo tanto los resultados son subjetivos. Al utilizar expertos en el tema se espera que este mismo estudio en otro contexto fuera de la ciudad de Barranquilla arroje resultados similares, a pesar que es una opinión subjetiva.

## **1.2 Trabajos futuros**

Como trabajo futuro se propone desarrollar e implementar un prototipo de estimación de costos de software, para validar este modelo en un ambiente real.

Desarrollar un análisis sobre el impacto financiero al aplicar el modelo con el fin de verificar el mejoramiento de los procesos para la estimación de costos de software.

### 1.3 Referencias Bibliográficas

- Addison, T. (2002). "Controlling Software Project Risks - an Empirical Study of Methods used by Experienced Project Managers." SAICSIT: 128 – 140.
- Adriana Gómez, M. d. C. L. y A. O. Silvina Migani (2000). "UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE."
- Al-Sakran, H. (2006). "Software Cost Estimation Model Based on Integration of Multi-agent and Case-Based Reasoning." Journal of Computer Science **2**: 276-282.
- Alberto Cueto Vigil. (2012). "de la I Encuesta de GP de TI en Colombia. V Jornada de Gerencia de Proyectos de TI.", from [http://www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/X\\_Jornada\\_Gerencia/ConferenciaAlbertoCuetoVigil.pdf](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/X_Jornada_Gerencia/ConferenciaAlbertoCuetoVigil.pdf).
- Albrecht, A. J. y J. E. Gaffney (1983). "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation." Software Engineering, IEEE Transactions on **SE-9**(6): 639-648.
- Álvarez Sáez, J. (2009). Colección de Apuntes Asignatura "Planificación de Sistemas informáticos", Escuela Politécnica Superior de Jaén.
- Bajaj, N. et al. (2006). "Software Estimation – A Fuzzy Approach." ACM SIGSOFT Software Engineering Notes **31** 5.
- Barry W, B. et al. (2000). "Software Cost Estimation with COCOMO II". NJ, USA, Prentice Hall.
- Boehm, B. et al. (1995). "Cost Models for Future Software Life Cycle Processes: COCOMO II." ANNALS OF SOFTWARE ENGINEERING **1**: 57-94.
- Boehm, B. W. (1981). Software Engineering Economics, Prentice Hall PTR Upper Saddle River.
- Boehm, B. W. (1984). "Software engineering economics." IEEE Trans. on Soft. Eng. **10**: 7-19.
- Bozhikova, V. y M. Stoeva (2010). An Approach for Software Cost Estimation. International Conference on Computer Systems and Technologies, CompSysTech'10.
- Capers, J. (2006). "Software Cost Estimating Methods For Large Projects " Chief Scientist and Founder Software Productivity Research **5**: 13.
- Comunicaciones, M. d. (2008). Plan Nacional de Tecnologías de la Información y las Comunicaciones. Bogota, Colombia, Ministerio de Comunicaciones: 78.
- Chaos (2001). EXTREME CHAOS REPORT.
- Chiu, N.-H. y S.-J. Huang (2006). "Optimization of analogy weights by genetic algorithm for software effort estimation." scientDirect **48**: 1034–1045.
- Chiu, N.-H. y S.-J. Huang (2007). "The adjusted analogy-based software effort estimation based on similarity distances." Journal of Systems and Software **80**(4): 628-640.
- Da Yang, Q. W. et al. (2008). "A Survey on Software Cost Estimation in the Chinese Software Industry." ESEM '08: 253-262
- Dalkey, N. C. et al. (1999). "The Delphi Method, III: Use of self rating to improve group estimates." Technological Forecasting and Social Change **1**: 283-291.
- Durán Rubio, S. E. (2003). "Puntos por Función. Una métrica estándar para establecer el tamaño del software." Boletín de política informática **6**: 8-16.
- Farrish, J. (2007). The relationship between project characteristics and the expert estimation of software development and maintenance. 1448395 M.S., University of Nevada, Las Vegas.
- Felix, C. E. W. a. C. P. (1997). "A method of programming measurement and estimation." **16**: 54-73.
- Forigua, S. P. y O. A. Ballesteros (2007). Propuesta de un modelo de análisis para estimación del tamaño del software y gestión de costos y riesgos a partir de requerimientos funcionales, PONTIFICIA UNIVERSIDAD JAVERIANA.
- Galbraith, J. R. (1974). "Organization Design: An Information Processing View." Interfaces **4**: 28-36.



- Genshowsky, G. et al. (2005). Gestión de proyectos de software. U. T. F. S. María: 187.
- Gordon Xu, J. A. G. (2006). "An Exploratory Study of Killer Applications and Critical Success Factors in M-Commerce." Journal of Electronic Commerce in Organizations: 63-79.
- Gray, A. R. et al. (1999). Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgment. Software Metrics Symposium, 1999. Proceedings. Sixth International.
- Gregory J. Skulmoski, F. T. H., Jennifer Krahn (2000). "The Delphi Method for Graduate Research." Journal of Information Technology Education **6**.
- Grimstad, S. et al. (2006). "Software effort estimation terminology: The tower of Babel." Information and Software Technology **48**(4): 302-310.
- Group, S. (2004). 2004 third quarter research report.
- Hareton Leung, Z. F. (2002). "Software Cost Estimation." Department of Computing The Hong Kong Polytechnic University.
- Heemstra, F. J. (1990). "Software cost estimation." Butterworth-Heinemann Ltd **34**: 286 - 297.
- Heemstra, F. J. y R. J. Kusters (1991). "Function point analysis:evaluation of a software cost estimation model." European Journal of Information Systems **4**: 229-231.
- Heemstra, F. J. y R. J. Kusters (1988). A Conceptual Framework for Software Cost Control and Estimation. Eindhoven, Netherlands, Industrial Engineering and management science.
- Hill, J. et al. (2000). "Experts' estimates of task durations in software development projects." International Journal of Project Management **18**(1): 13-21.
- Hughes, B. y M. Cotterell (2004). Software Project Management.
- Hughes, R. T. (1996). "Expert judgement as an estimating method." Information and Software Technology **67-75**.
- Jeffery, R. et al. (2000). "A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data." Information and Software Technology **42**(14): 1009-1016.
- Jorgensen, M. (2005). "Practical Guidelines for Expert-Judgment-Based Software Effort Estimation." IEEE Software **22**(3): 57-63.
- Jørgensen, M. (1997). An empirical evaluation of the MkII FPA estimation model. Norwegian Informatics Conference. Voss, Norway, Tapir, Oslo: 7-18.
- Jørgensen, M. (2004). "A review of studies on expert estimation of software development effort." Journal of Systems and Software **70**(1-2): 37-60.
- Jørgensen, M. (2004). "Top-down and bottom-up expert estimation of software development effort. Information and Software Technology." Information & Software Technology - INFISOFT **46**: 3-16.
- Jørgensen, M. y D. I. K. Sjøberg (2003). "An effort prediction interval approach based on the empirical distribution of previous estimation accuracy." Information and Software Technology **45**(3): 123-136.
- Joseba Esteban López, J. J. D. C. (2008). "Estimación del esfuerzo Software: Factores vinculados a la aplicación a desarrollar." SISTEDES **2**: 36.
- Kasunic, M. (2005). Designing an Effective Survey. C. M. University.
- Kavantzias, N. et al. (2004). "Web Services Choreography Description Language (WS-CDL) version 1.0."
- Kavantzias, N. et al. (2004). "Web Services Choreography Description Language (WS-CDL) version 1."
- Kolodner, J. L. (1988). Case-Based Reasoning Workshop. San Mateo, CA., Morgan Kaufmann Publishers, Inc.
- Laird Linda M, B. C., Brennan M (2006). Software Measurement and Estimation a practical Approach., John Wiley & Sons.
- Lederer, A. L. y J. Prasad (1992). Nine management guidelines for better cost estimating, Communications of the ACM. **35**: 51 - 59

- Li, Y. F. et al. (2007). A study of genetic algorithm for project selection for analogy based software cost estimation. Industrial Engineering and Engineering Management, 2007 IEEE International Conference on.
- Li, Y. F. et al. (2009). "A study of project selection and feature weighting for analogy based software cost estimation." Journal of Systems and Software **82**(2): 241-252.
- Linstone, H. A. y M. Turoff (1975). "The Delphi Method Techniques and Applications." Addison-Wesley **200**: 6118.
- Madachy, J. R. (1997). "Heuristic Risk Assessment Using Cost Factors, IEEE Software." 51-59.
- Magazinius, A. y R. Feldt (2010). "Exploring the Human and Organizational Aspects of Software Cost Estimation." ACM SIGSOFT Software Engineering Notes.
- Mario Piattini V, F. G. R. (2008). Medición y estimación del software. Técnicas y métodos para mejorar la calidad y la productividad.
- Martin Shepperd, M. J. (2007). "A Systematic Review of Software Development Cost Estimation Studies." IEEE Transactions on Software Engineering **33**(1): 33-53.
- McAulay, K. (1987). Information Systems Development and the Changing Role of MIS in the Organisation. First New Zealand MIS Management Conference.
- McConnell, S. (2006). Software Estimation: Demystifying the Black Art, Microsoft Press A Division of Microsoft Corporation.
- Mittas, N. y L. Angelis (2008). "Combining Regression and Estimation by Analogy in a Semi-parametric Model for Software Cost Estimation." ESEM'08: 70-78.
- Moløkken-Østvold, K. et al. (2004). A Survey on Software Estimation in the Norwegian Industry. Proceedings of the 10th International Symposium on Software Metrics. I. C. SOCIETY. Blindern, 0316 Oslo, Norway.
- Moreno, A. M. (1995). Estimación de proyectos de software. España, Centro de estudios financieros y Facultad de Informatica (UPM).
- Paynter, J. (1996). Software Engineering: Education and Practice, 1996. Proceedings. International Conference. International Conference: 150-159
- Pendharkar, P. et al. (2003). A Probabilistic Model for Predicting Software Development Effort. Computational Science and Its Applications — ICCSA 2003. V. Kumar, M. Gavrilova, C. Tan y P. L'Ecuyer, Springer Berlin Heidelberg. **2668**: 581-588.
- Peters, K. (1999). "Software Project Estimation." Software Productivity Center Inc.
- Pow-Sang Portillo, J. A. I. P., Ricardo (2005). "Estimación y Planificación de Proyectos Software con Ciclo de Vida Iterativo-Incremental y empleo de Casos de Uso." 6.
- Pressman, R. S. (2002). Ingeniería del Software – Un Enfoque Práctico.
- Pressman, R. S. (2005). Ingeniería de Software. Un enfoque practico, Mc Graw Hill.
- Project Management Institute, I. (2008). Guía de los Fundamentos de la Dirección de Proyectos. Newtown Square, Pennsylvania, PMI Publications.
- Puertas Del Castillo, E. (2011). Método de integración empresarial orientada a servicios: Pequeñas y medianas empresas Investigativa, Universidad Tecnologica de Bolivar.
- Reich, C. S. (2009). "PRONÓSTICOS CUALITATIVOS." Retrieved 27 Febrero, 2013, from <http://shreich.wordpress.com/2009/10/07/pronosticos-cualitativos/>.
- Shepperd, M. (May 2007). Software project economics: a roadmap. Future of Software Engineering, Conference on Software Engineering: 304–315.
- Shepperd, M. y C. Schofield (1997). "Estimating software project effort using analogies." Software Engineering, IEEE Transactions on **23**(11): 736-743.
- Skulmoski, G. J. et al. (2000). "The Delphi Method for Graduate Research." Journal of Information Technology Education vol 6
- Sommerville, I. (2005). Ingeniería del Software, Pearson.
- Symons, C. R. (1988). "Function point Analysis: Difficulties and Improvements." IEEE Transactions on Software Engineering **14**(1).

- Wijayasiriwardhane, T. et al. (2011). "Effort estimation of component-based software development – a survey." IET Software **5**(2): 216.
- Wydenbach, G. a. P., J. (1995). "Software Project Estimation: a Survey of Practices in New Zealand." New Zealand Journal of Computing **6**: 317-327.
- Xu, G. y J. A. Gutiérrez (2006 ). "An Exploratory Study of Killer Applications and Critical Success Factors in M-Commerce." Journal of Electronic Commerce in Organizations: 63-79.
- Zheng, Y. et al. (2006). "An Extension of COCOMO II for the B-Method." EDSER'06: 11-14.
- Zulkefli B in Mansor, Z. M. K., N oor Habibah Hj Arshad, Saadiah Yahya (2010). "E-Cost Estimation Using Expert Judgment and COCOMO II." IEEE.

# **ANEXOS**

**GLOSARIO DE TERMINOS**

**Sprint:** Organización del desarrollo sobre los que se definen objetivos y se analizan, planifican y desarrollan tareas que cumplen con los requisitos establecidos con el cliente en este periodo de tiempo.

**Source control:** La gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

**Branching:** Control de la revisión y la gestión de configuración de software, es la duplicación de un objeto bajo control de revisión (tal como un código fuente de archivo, o un árbol de directorios), de modo que las modificaciones pueden ocurrir en paralelo a lo largo de las dos ramas.

**Enviroments:** proporciona servicios integrales para los programadores informáticos para el desarrollo de software. Una IDE normalmente consiste en un editor de código fuente, construir automatización herramientas y un depurador.

**QA:** consiste en un medio de control de los de ingeniería de software procesos y métodos utilizados para asegurar la calidad. Los métodos por los cuales esto se logra son muchas y variadas, y pueden incluir la garantía de la conformidad con una o más normas, tales como la norma ISO 9000 o un modelo como CMMI.

## **Anexo A- Perfil de los panelistas**

A continuación se muestra un breve resumen de cada uno de los panelistas que respondieron por lo menos dos rondas, así como otras personas que participaron:

Experto	Nombre	Empresa	Formación Académica	Experiencia	Número de empleados
1	Luz Mery Bermejo	Extreme technologies s.a	especialista en sistemas de información	9 años con experiencia de ingeniera de software y gestión de proyectos. Actualmente se desempeña como Gerente de Ingeniería	15
2	Roberto Morales	fundación I+D+I	Ingeniero de Sistemas	Investigador, enfocado en el manejo, diseño y creación de proyectos de Investigación empleando software de escritorio y tecnología web en lenguaje Java, .Net, JavaScript, Extjs, PHP y ASP, Joomla como sistema gestor de contenido y codeigniter como framework php, con conocimientos en mantenimiento de equipos y redes de computadores,. Actualmente se desempeña como director de desarrollo de software en la empresa I+D+I.	10
3	Nicolás costa	Koombea	Ingeniero de Sistemas	Empezó en la empresa koombea con el cargo de Bussines Manager donde se encargó de liderar el proceso de crecimiento fortaleciendo las bases de la compañía y apoyando el trabajo en equipo. Actualmente es mentor experto juez del programa Appsco liderado por MinTIC para la etapa de Ideación, prototipado y validación; así como asesor de procesos de internacionalización para emprendimientos de IT en etapa de aceleración.	40
4	Néstor Alvarado Namen	Nansoft technology s.a	Ingeniero de Sistema	Conocimientos en proyectos de software y manejo de herramientas para la gestión de proyectos de software.	
5	Juan Carlos Rúgeles	ASP solutions	Ingeniero de Sistemas	Fundador y gerente comercial de la empresa ASP solutions s.a	10
6	Monica Soto	Geotech	Ingeniero de Sistemas, Magister en Gobierno de Tecnologías Informáticas de la Universidad	Con amplia experiencia en la dirección y ejecución de proyectos de tecnología y desarrollo de software, y en procesos de implementación de sistemas de gestión de calidad.	15

			del Norte de Barranquilla		
7	Hernán Pájaro	Technet Ltda	Ingeniero de Sistemas	gerente de operaciones de la empresa Integranet Ltda y gerente general de la empresa Technet Ltda. Scrum Master en proyectos de desarrollo a la medida en plataforma Windows y WEB, Arquitecto de Soluciones IT.	8
8	Ricardo Rivera	VSI nearshore Outsourcing	Ingeniero de Sistemas	con 14 años de experiencia en la industria de software Liderazgo Empresarial, Nearshore Outsourcing, Ventas Estratégicas, Gestión Total de la Calidad, Análisis de Datos de Proyectos y Gestión de Producto, SDLC, Coaching, enlace con el cliente, Branding, Experiencia de Usuario, Diseño y Gestión, grupos de enfoque, Marketing Online. Se desempeña como SVP, Technology & Innovation en la empresa VSI Nearshore Outsourcing	40
9	Kenny Quevedo Doria	Consaware	Ingeniero de Sistemas	Especialista en desarrollos web utilizando tecnologías Microsoft, desarrollo de aplicaciones tipo escritorio, desarrollo de bases de datos utilizando motor SQL Server. Certificación Microsoft en NET Framework 4, Web Applications, .NET Framework 4, Service Communication Applications, .NET Framework 2.0, Web-based Client Development. Coordinador de proyectos Tecnológicos de la empresa Conseware.	19
10	Jacobo Llanos Cerro	manejo software avanzado	Ingeniero de Sistemas	Gerente general de la empresa compañía líder en software de manejo avanzado Ltda.	8

## **Anexo B – Instrucciones de la encuesta ronda 1**

En esta sección las instrucciones completas a los panelistas para cada una de las tres rondas del proceso de Delphi se incluyen en este apéndice.

### **RONDA 1**

#### **INTRODUCCIÓN**

Los diferentes métodos de estimación de costos de software permiten realizar estimaciones más cercanas a la realidad, donde las compañías adquieren una mejor planificación y pueden utilizar mejor sus recursos, siendo cada día más relevante dentro de la Ingeniería de software. Al revisar la literatura y estudios a nivel mundial, se han identificado escenarios generales que establecen los aspectos a tener en cuenta para realiza estimaciones más precisas al momento de realizar un proyecto de Desarrollo de software.

El software es uno de los componentes más complejo y costoso de fabricar en los sistemas de información que se utilizan actualmente. Para sistemas complejos y personalizados, un error en la estimación de costo puede marcar la diferencia entre beneficios y pérdidas para una compañía, ya que una de las características que debe tener un producto de software es que su costo sea adecuado, de lo contrario el proyecto puede fracasar.

Para los escenarios que se describen a continuación, por favor revise cada uno de ellos y adicione comentarios y/o sugerencias, si los tiene, de manera clara; puede usted agregar, quitar características, elementos u otros a los escenarios descritos y/o proponer otro que no se mencionen en este documento.

#### **LISTA DE ESCENARIOS**

##### **Escenario A:**

Las empresas desarrolladoras de software solicitan el apoyo de la experiencia de varias personas que están familiarizadas con el desarrollo de aplicaciones de software similares. La característica principal en la empresa es la ausencia de datos cuantificados de proyectos anteriores. Cada una de las personas estima el costo del proyecto de desarrollo de software basándose solo en su experiencia y conocimientos anteriores; luego las estimaciones se comparan y con todo el grupo se discuten las diferencias e inconsistencias de la siguiente manera:

1. El coordinador presenta a cada persona, y a cada uno le entrega un formulario con especificaciones para registrar las estimaciones.
2. Cada persona lo diligencia en forma individual (sin discutir con los demás) y se permiten hacer las preguntas al coordinador.



3. El coordinador prepara un resumen de todas las estimaciones de los expertos, se incluye la mediana de un formulario para solicitar otra iteración de las estimaciones de los expertos y la justificación para estas.
4. Si no se llega a un consenso se repite el paso 2.
5. Tantas rondas como sea apropiado hasta alcanzar un consenso.

El inconveniente que se presenta es que es difícil para las personas que intervienen en el proyecto, reproducir y utilizar la experiencia en un futuro ya que consideran que no hay un proceso estructurado de cómo se producen las cifras para realizar los cálculos de las estimaciones de costos del proyecto de software. En la medida en que se ejecuta el proyecto de software se pueden comprobar las estimaciones obtenidas y si se da el caso volver a estimar. La empresa considera entre las ventajas de este método, es relativamente barato y se puede obtener bastante precisión si los expertos tienen experiencia directa en sistemas similares; pero a la vez puede ser muy impreciso sino se dispone de expertos familiarizados con el área del proyecto, así como olvidar factores importantes del nuevo proyecto, creyendo que es casi igual al anterior.

### **Escenario B:**

Las empresas desarrolladoras de software cuentan con datos cuantificables y/o históricos de proyectos anteriores, utilizados para predecir el costo del nuevo proyecto. La empresa utiliza los valores de parámetros como el alcance, el costo, el presupuesto y la duración, o medidas de escala tales como el tamaño, el peso y la complejidad de un proyecto anterior similar, como base para estimar el mismo parámetro o medida para un proyecto actual. De esta manera, el líder del proyecto estima las similitudes entre el nuevo proyecto y los proyectos anteriores y se escoge la más parecida. Existen multitud de medidas de similitud entre ejemplares, siendo las más usadas las distancias de Euclides, de Manhattan y de Minkowski. La más utilizada es la de Euclides:

$$\text{Distancia } (P_x, P_y) = \sqrt{\sum_{i=1}^m (P_{xi} - P_{yi})^2}$$

Con esta ecuación se calcula el esfuerzo hallado en cada uno de los proyectos. Si la distancia entre los dos proyectos es 0, se deduce que los valores de la función de los esfuerzos calculados son los mismos. Sin embargo la distancia puede aumentar a medida que disminuye la similitud; la distancia métrica mide la similitud entre dos proyectos en términos de manejador de esfuerzo y la suma de los cuadrados del esfuerzo es la distancia total entre los proyectos. Por lo anterior, el análogo más cercano al proyecto que se estima es el proyecto con la distancia mínima euclidiana entre todos los proyectos históricos. Pero el problema es que la empresa necesita constantemente un mantenimiento sistemático de la base de datos histórica y los expertos la consideran menos exactas en cuanto a las estimaciones.

### **Escenario C:**

La empresa desarrolladora de software considera que el costo del proyecto del software está en función de lo que el cliente está dispuesto a pagar ya que no se puede dar el lujo de perder un cliente. El problema que se presenta es que la probabilidad de que el cliente obtenga el producto que quiere es pequeña, ya que los costes no reflejan realmente el trabajo requerido para su desarrollo y se puede

generar retrasos en la entrega u obligar al equipo de desarrollo a trabajar horas extras. Las estimaciones de costos se basan en el presupuesto del cliente en lugar de la funcionalidad del software.

### **Escenario D:**

La empresa desarrolladora de software no se complica realizando cálculos manuales y decide adquirir una herramienta para realizar sus estimaciones. Estos software son comerciales y proporcionar un núcleo de funciones, incluyendo entre sus atributos:

- Lógica de dimensionamiento para especificaciones, código fuente y casos de evaluación
  - Nivel de fase, nivel de actividad, y estimación de nivel de tarea.
  - Ajustes para períodos de trabajo específicos, vacaciones y horas extraordinarias.
  - Ajustes para salarios locales e índices de carga.
  - Ajustes para varios proyectos de software como militares, sistemas, comerciales, etc.
  - Apoyo para métricas de puntos de función, métricas de líneas de códigos o ambas.
  - Apoyo para “backfiring”, es decir, convertir líneas de códigos a puntos de función
  - Apoyo tanto para nuevos proyectos como a proyectos de mejora y mantenimiento.
  - Medios de medición para reunir datos históricos
  - Costo y tiempo para completar estimaciones que combinan datos históricos con datos proyectados
  - Conversión monetaria para acordar proyectos en el exterior
- 
- \* Es importante calcular para todos los escenarios el tamaño del producto (en número de líneas de código fuente o puntos de función), estimar el esfuerzo (personas-mes) es fácil para la empresa si se tiene una estimación exacta del tamaño y los datos previos de la organización en proyectos similares para estimar la duración total del proyecto. Es importante conocer como su compañía realiza estos calculo, puede anexarlos al documento.

## **Anexo C –Instrucciones de la encuesta ronda 2**

### **INTRODUCCIÓN**

Clasificar y Valorar las listas de tendencias y escenarios para la Estimación de costos de proyectos para las empresas desarrolladoras de software.

La lista de escenarios presentada en la primera ronda, son presentados con los elementos adicionados y recomendaciones hechas por el panel de expertos. Son 5 posibles tendencias o escenarios que afectan de una exitosa Estimación de costos de proyectos para las empresas desarrolladoras de software.

El propósito de esta ronda para el panel de expertos es de manera individual:

1. Clasificar todos los elementos en términos de importancia para el éxito de la Estimación de costos de proyectos para las empresas desarrolladoras de software.
2. Valorar las tendencias y los escenarios enumerados en términos de probabilidad ocurrencia.
1. Proveer cualquier información adicional.

Los resultados de esta ronda serán analizados por mi persona, después se devolverán los resultados al panel de expertos para la tercera y última ronda, así que por favor sea claro en su razonamiento después de la puntuación de los escenarios.

#### Instrucciones para la clasificación y Valoración

En primer lugar, la lista tiene una clasificación de 5 elementos en orden de importancia donde se clasificarán todos los elementos en términos de importancia para el éxito de la Estimación de costos de proyectos para las empresas desarrolladoras de software.

En segundo lugar, el porcentaje de probabilidad según las ocurrencias teniendo de referencia la siguiente escala:

#### Significado de la clasificación

5. Esto definitivamente va a suceder en un futuro próximo (6-24 meses).
4. Esto probablemente sucederá en un futuro próximo a mediano plazo.
3. Esto puede suceder en el futuro o puede también no ocurrir.
2. Esto es poco probable que suceda en el futuro.
1. Esto definitivamente no ocurrió.

Por último, proporcionar cualquier comentario adicional sobre cada escenario que desea usted ofrecer.

Si usted tiene alguna pregunta no dude en contactarme, ya sea por correo o por teléfono.

Nohora Mercado Caruso

[Nmercado1@cuc.edu.co](mailto:Nmercado1@cuc.edu.co)

3005674999-3362246

## RONDA 2

Con los escenarios consultados a nivel mundial se encontró que en el contexto actual la tendencia de las estimaciones son parecidas pero con algunas modificaciones. De esta manera se identifican características similares en los nuevos escenarios propuestos por las empresas:

### Escenario Empresa A1:

Una de las formas de realizar la estimación de esfuerzo o tamaño relativo de las tareas de desarrollo es basarse en utilizar una baraja de cartas que se encuentran enumeradas mostrando la secuencia Fibonacci: 0, 1/2, 1, 2, 3, 5, 8, 13, 20, 40, 100. Se utiliza esta secuencia para identificar la incertidumbre de cada una de las estimaciones. Cada uno de los integrantes del proyecto posee un mazo de cartas; el moderador que puede ir guiado por el coordinador del proyecto proporciona las características de la tarea a desarrollar, y el grupo puede intervenir para aclarar sus dudas.

Cada persona coloca su carta boca abajo, de forma simultánea que representa la estimación para una tarea. Con la serie Fibonacci se define que representa las unidades de las cartas, ya sean días, u horas.

Las personas que saquen números alejados del promedio pueden dar su justificación, y se repite el proceso hasta que se llegue a un consenso. Después de esto se sigue el mismo proceso pero con otra tarea.

Clasificar este tema (Escenario A1 de estimación de costos de software)		De 5
---	--	------

Valoración de este tema :	5	4	3	2	1
Resaltar la puntuación	Totalmente de acuerdo	Parcialmente de acuerdo	Ni en acuerdo/ ni en desacuerdo	Parcialmente en desacuerdo	Totalmente en desacuerdo

### Escenario Empresa A2:

La empresa considera que es importante crear una base de datos histórica para medir el costo de un nuevo proyecto. La base de conocimiento tendría información como descripción del requerimiento, complejidad del requerimiento, tiempo estimado, tiempo real, duración general del proyecto en horas, valor por hora dependiendo del perfil del desarrollador, etc.

En cada estimación lo importante es disminuir el factor de incertidumbre. Si el desarrollador tiene certeza de los desarrollos que hay que realizar y su complejidad puede estimar las horas que se requieren y los tiempos de cada etapa, lo que va a dar un alto nivel de precisión en el costo. Este tipo de información histórica genera estadísticas del tiempo para realizar una tarea y es el referente para nuevas estimaciones.

Clasificar este tema (Escenario A2 de estimación de costos de software)		De 5
---	--	------

Valoración de este tema :	5	4	3	2	1
Resaltar la puntuación	Totalmente de acuerdo	Parcialmente de acuerdo	Ni en acuerdo/ ni en desacuerdo	Parcialmente en desacuerdo	Totalmente en desacuerdo

**Observaciones:**

--

**Escenario Empresa A3:**

La empresa debe establecer el alcance y los requisitos del proyecto en cada una de las fases para realizar una buena estimación. Por cada una de las fases se descomponen los requisitos en tareas o funcionalidades más específicas y sobre estas se realizan estimaciones de tiempo, así como el valor por hora de desarrollo dependiendo del perfil del desarrollador que la va a realizar.

Se establece el tiempo de entrega con el cliente y si el cliente decide que el valor establecido es muy alto se realizan negociaciones y se eliminan funcionalidades hasta llegar a un valor en que se sienta satisfecho, esto normalmente se hace cuando la empresa cuenta con un producto que comercializa.

Clasificar este tema (Escenario A3 de estimación de costos de software)		De 5
---	--	------

Valoración de este tema :	5	4	3	2	1
Resaltar la puntuación	Totalmente de acuerdo	Parcialmente de acuerdo	Ni en acuerdo/ ni en desacuerdo	Parcialmente en desacuerdo	Totalmente en desacuerdo

**Observaciones:**

--

**Escenario Empresa A4**

El cliente se considera con poca fundamentación para establecer requerimientos, solo cuenta con una idea general del producto que quiere. La empresa desarrolladora genera un producto mínimo viable. Se realizan

entregas parciales al cliente (pueden ser semanales) el cliente lo prueba y con las recomendaciones se agregan funcionalidades y modificaciones hasta llegar al producto deseado.

En el ciclo de vida de desarrollo se cuenta con la participación activa del cliente y al desarrollo incremental con interacciones muy cortas. Una de las características es que los requisitos son cambiantes, por lo que el cliente puede ver en tiempo real las actividades que está desarrollando el grupo de desarrollo, así como el número de horas que llevará e n culminarse dicha actividad.

Para realizar el seguimiento de las tareas el coordinador del proyecto se reúne con su grupo de trabajo y se verifica el estado de los Sprint. Es importante establecer prioridad en las tareas ya que le cliente verifica y prueba las funciones básicas, se realiza un feedback para identificar las mejoras.

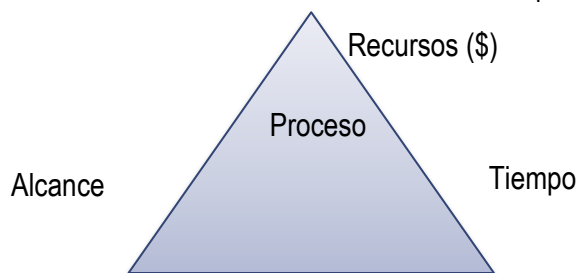
Clasificar este tema (Escenario A4 de estimación de costos de software)		De 5
---	--	------

Valoración de este tema :	5	4	3	2	1
Resaltar la puntuación	Totalmente de acuerdo	Parcialmente de acuerdo	Ni en acuerdo/ ni en desacuerdo	Parcialmente en desacuerdo	Totalmente en desacuerdo

**Observaciones:**

**Escenario Empresa A5:**

Se debe tener en cuenta el alcance del proyecto para realizar una buena estimación del costo de este. En muchas ocasiones existe un problema de aprendizaje donde el líder del proyecto y desarrolladores no entiende lo que desea el cliente. Es importante primero conocer cómo opera el cliente, para lo cual se deben realizar reuniones con el cliente. Al determinar cómo se va a hacer, se determina el tiempo, el cual impacta en los recursos.



Para determinar el alcance, se tiene en cuenta los requisitos en cada una de las fases del software:

1. Análisis
2. Diseño
3. Desarrollo
4. Pruebas
5. Despliegue de entregas

Se utiliza el siguiente cuadro para estimar los costos, teniendo en cuenta que cada persona trabaja cuatro horas diarias. Se descompone en funcionalidad o actividades (con la lista anterior), siendo mucho más fácil estimar el tiempo de desarrollo del proyecto.

<b>Código</b>	<b>Funcionalidad/Actividad</b>	<b>Horas</b>
C1	Horas Efectivas	
C2	Eficiencia	
.	Horas disponibles	Dividir horas efectivas/eficiencia
.	Horas semanales	
.	Tiempo de entrega por semanas	
.	Valor hora Base	
.	Costos de software	

El proyecto avanza por medio de iteraciones con el cliente que se denominan sprint, cada Sprint tiene una duración de 1 a 4 semanas.

La empresa con su mayoría de clientes firma un contrato de soporte, en el cual constantemente está realizando mantenimiento al software según las nuevas necesidades o reportes que el cliente necesite.

Clasificar este tema (Escenario A5 de estimación de costos de software)		De 5
---	--	------

<b>Valoración de este tema :</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Resaltar la puntuación	Totalmente de acuerdo	Parcialmente de acuerdo	Ni en acuerdo/ ni en desacuerdo	Parcialmente en desacuerdo	Totalmente en desacuerdo

**Observaciones:**

--

## Anexo D –Ejemplo de envío de correo a los panelistas

**De:** Nicolas Costa [mailto:nicolas.costa@koombea.com]

**Enviado el:** viernes, 14 de junio de 2013 11:13 a.m.

**Para:** MERCADO CARUSO NOHORA NUBIA

**Asunto:** Re: Participación Koombea- investigación Maestría

2013/5/15 MERCADO CARUSO NOHORA NUBIA <[nmercado1@cuc.edu.co](mailto:nmercado1@cuc.edu.co)>

Barranquilla 15 de Mayo del 2013

Señores:

KOOMBEA

Ing. Nicolas Costa

Anteriormente manifestó su interés en participar en el proyecto de Investigación “MODELO DE ESTIMACIÓN DE COSTOS DE SOFTWARE PARA LAS EMPRESAS DESARROLLADORAS DE SOFTWARE” donde se evaluará como las empresas realizan las estimaciones de costos en cada uno de los diferentes proyectos de software. Cabe resaltar que al utilizar el método Delphi el proceso se lleva a cabo de una manera anónima, por lo que usted no conocerá cuales son las empresas participantes. Igualmente la información que se proporcione será fundamental para desarrollar el modelo que al final del estudio será compartido con todas las empresas participantes para que tengan una opción ser mas competitivos en el mercado.

De acuerdo con el método Delphi, estoy circulando un documento inicial que describe varios escenarios de cómo realizar estimaciones de costos en proyectos de desarrollo de Software, de los cual se espera que usted con su experiencia y conocimiento pueda revisar y sugerir mejoras sobre la descripción de los escenarios. Por favor, complete esta revisión y devolverlo con sus comentarios a:

[nmercado1@cuc.edu.co](mailto:nmercado1@cuc.edu.co)

El objetivo de este estudio es predecir la adopción de un enfoque orientado a como las empresas desarrolladoras de software realizan estimaciones de costos de software, con el fin de ser más competitivas en un contexto global. Favor revisar el documento adjunto donde podrá encontrar los diferentes escenarios y las instrucciones a seguir.

Agradecemos su valiosa colaboración y apoyo.

Cordialmente,

Nohora Mercado Caruso

Coordinadora de Practicas CUC

Facultad de Ingeniería

Grupo de Investigación Ing. software y Redes

[nmercado1@cuc.edu.co](mailto:nmercado1@cuc.edu.co)

3005674999



---

De: Nicolas Costa [[nicolas.costa@koombea.com](mailto:nicolas.costa@koombea.com)]

Enviado el: miércoles, 10 de abril de 2013 0:36

Para: MERCADO CARUSO NOHORA NUBIA

CC: [jhonatan@koombea.com](mailto:jhonatan@koombea.com)

Asunto: Re: Participación Koombea- investigación Maestría

Hola Nohora, quedo a la orden para atender a dicho panel. Por favor me confirmas las fechas para agendarlo.

2013/4/5 MERCADO CARUSO NOHORA NUBIA

<[nmercado1@cuc.edu.co](mailto:nmercado1@cuc.edu.co)<mailto:[nmercado1@cuc.edu.co](mailto:nmercado1@cuc.edu.co)>>

Barranquilla 5 Abril 2013

Señores:

KOOMBEA

Ing. Jonathan Tarud

Gerente

Cordial saludo. La presente es para solicitar de la manera más atenta su colaboración como experto panelistas o la recomendación de un investigador a un proyecto de investigación que estoy adelantando en la Maestría en Ingeniería con Énfasis en Sistemas y Computación de la Universidad Tecnológica de Bolívar. La investigación consiste en predecir la adopción de un enfoque para mejorar los procesos de estimación de costos de software en empresas desarrolladoras de software, que las hagan más competitivas en un contexto global, para esto se utilizará el "Método Delphi"

El método consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes.

Su colaboración o de la persona que me pueda recomendar sería como panelista en esta metodología.

Gracias por la atención prestada y espero su pronta respuesta.

Business Manager

Koombea

T: 360 5497

W: [www.koombea.com](http://www.koombea.com)