

## Mejora de los procesos de estimación de costos de software. Caso del sector de software de Barranquilla

Nohora Mercado Caruso\*  
Edwin Puerta del Castillo\*\*  
Katherinne Salas Navarro\*\*\*

Recibido: 28/10/2014 • Aceptado: 13/05/2015

### **Resumen**

El presente artículo es resultado de la investigación: “Diseño de un modelo para mejorar los procesos de estimación de costos para las empresas desarrolladoras de software”. Se presenta una revisión de la literatura con el fin de identificar tendencias y métodos para realizar estimaciones de costos de software más exactas. Por medio del método predictivo Delphi, expertos pertenecientes al sector de software de Barranquilla en Colombia, clasificaron y valoraron según la probabilidad de ocurrencia cinco escenarios realistas de estimaciones. Se diseñó un experimento completamente aleatorio cuyos resultados apuntaron a dos escenarios estadísticamente similares de manera cualitativa, con lo que se construyó un modelo de análisis basado en tres agentes: metodología, capacidad del equipo de trabajo y productos tecnológicos; cada uno con tres categorías de cumplimiento para lograr estimaciones más precisas.

*Palabras clave:* Estimación de costos, método Delphi, proyecto de software, escenarios, juicio experto, estimación por analogía, precio para ganar, gestión de software.

---

\* Maestría en Ingeniería. Énfasis en Ingeniería de sistemas. Docente Tiempo completo, Ingeniería Industrial, Universidad de la Costa CUC. Calle 58 # 55-66. Barranquilla, Colombia. nmercado1@cuc.edu.co

\*\* Maestría en Ingeniería. Énfasis en Sistemas. Docente Tiempo Completo, Ingeniería de Sistemas. Universidad Tecnológica de Bolívar UTB. Parque Industrial y Tecnológico Carlos Vélez Pombo, Km 1 Vía Turbaco. Cartagena. epuerta@unitecnologica.edu.co

\*\*\* Maestría en Ingeniería. Énfasis en Ingeniería Industrial. Docente Tiempo completo, Ingeniería Industrial, Universidad de la Costa CUC. Calle 58 # 55-66. Barranquilla, Colombia. Ksalas2@cuc.edu.co

## Improvement of software process estimation. Applied case of software area in Barranquilla

### Abstract

Summary-The present paper is the result of the following investigation: “Design of a model to improve the cost estimation processes for software developing companies”. It is presented a revision of the literature at an international level for identifying tendencies and methods for more accurate software cost estimations. Through the Delphi predictive method, a group of experts in the field of software development in Barranquilla qualified and assessed from the probability of occurrence five realistic estimation scenarios. A completely random experiment was designed whose results point to two scenarios statistically similar in a qualitative manner, from that a three agents model was built: Methodology, capacity of the work team and technological products; each of those with three categories of fulfilment to achieve more precise estimations.

*Key words:* Cost estimation, Delphi method, software project, scenarios, expert judgment, estimation by analogy, price to win, software management.

## INTRODUCCIÓN

La complejidad en los desarrollos de software se denominó 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 de software [1]. Es por esto que 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 de proyectos 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 y una estimación es una predicción de cuánto tiempo durará o costará un proyecto [2]. 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.

Las empresas pueden esforzarse en desarrollar proyectos de software competitivos, pero existe la necesidad de medir y controlar este esfuerzo para no resultar en pérdida de dinero, tiempo, empleados, etc. 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. 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 [3].

Es por esto que se propone un modelo de análisis que permite mejorar los procesos de estimaciones de costos en las empresas desarrolladoras, teniendo en cuenta que lo que garantiza que un proyecto sea exitoso es la realización de estimaciones adecuadas sobre el esfuerzo traducido en la unidad de tiempo hombre.

### 1. LA IMPORTANCIA DE REALIZAR ESTIMACIONES DE COSTOS DE SOFTWARE

La necesidad de estimar se evidencia en un informe estadístico en Canadá “EXTREME CHAOS REPORT” [4]. 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 su reputación.

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. 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. Por lo anterior, 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 [5]. Existen muchos métodos para realizar estimaciones, y a través del tiempo su uso se ha venido intensificando. En la tabla 1 se pueden detallar, teniendo en cuenta sus principales ventajas y limitaciones.

Tabla 1. Ventajas y desventajas de los métodos de estimación

<i>Método de Estimación</i>	<i>Ventajas</i>	<i>Limitaciones</i>
COCOMO II	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.

Fuente: elaboración propia

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 con lo presupuestado [6]. Por todo lo anterior, la estimación de costos de software es importante para la planificación, programación, presupuesto y establecimiento del precio indicado al desarrollo del software [7]. Otros

autores como Pendharkar [8] y Martin Shepperd [9] 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.

## 2. MATERIALES Y MÉTODOS

Se utilizó el método Delphi para identificar las tendencias o escenarios de cómo las empresas pueden llevar a cabo sus procesos de estimación de costos. Este 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, con el objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. Según [10-12], el método Delphi se basa en una serie de fases que consisten en

Fase 1: Formulación del problema.

Fase 2: Elección de expertos.

Fase 3: Elaboración y lanzamiento de los cuestionarios.

Fase 4: Desarrollo práctico y explotación de resultados.

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, las cuales tuvieron una duración de 3 meses.

Cabe resaltar que 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», además de 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” de 1975 como guía a algunos autores [11, 12]. El diseño del formato es texto simple y utiliza un formato sencillo para incentivar a su diligenciamiento.

La lista de temas para realizar la valoración y clasificación proviene de la revisión de la literatura sobre métodos de estimación de costos de software. Se presentaron las siguientes 4 categorías:

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

### 3. RESULTADOS

Para analizar los datos obtenidos de las repuestas de clasificación y valoración ofrecida por parte de los expertos, se hizo un análisis estadístico por medio de un diseño de experimento aleatorizado. Este análisis arrojó un resultado cuantitativo de las tendencias más significativas para los panelistas. De esta manera los escenarios escogidos por los expertos son más representativos estadísticamente y sirvieron de base para el diseño del modelo propuesto. Cabe resaltar que no se describe a cuál método corresponde cada escenario, ya que *Delphi* recomienda no especificarlo con el fin de no generar sesgo en la etapa de reconocimiento.

En la primera ronda los expertos hicieron comentarios según su experiencia en los procesos de estimación de costos de software. De esta manera se identificaron cambios en los escenarios planteados. Los escenarios se formularon al consultar la revisión bibliográfica sobre estudios en el mundo sobre cuáles son los métodos de estimación más usados. En la segunda ronda se mejoraron y eliminaron los escenarios que no estaban acordes con 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. El objetivo fue obtener una calificación en la probabilidad de ocurrencia en los diferentes escenarios y analizar los resultados, para determinar áreas de acuerdo y contención (ver tabla 1).

En la tercera ronda se verificó 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 hace ajustes de clasificación y valoración, y tiene la opción de responder a comentarios realizados por otros panelistas.

#### 3.1 Análisis de los resultados

Para analizar los datos obtenidos, se utiliza un diseño experimental de las repuestas de clasificación y valoración, ofrecidas por parte de los expertos. Este análisis arroja

un resultado cualitativo de las tendencias o escenarios para tomar decisiones sobre cuáles métodos son los más significativos para los panelistas.

a. Definición del problema y factores de clasificación

En la tabla 2 se pueden apreciar 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 2. Respuesta de experto factor de clasificación en los escenarios

<i>Expertos</i>	<i>Escenarios</i>				
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
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

Fuente: elaboración propia

**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

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

**Número 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 observa 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 (ver tabla 3).

Tabla 3. Resumen estadístico para la clasificación

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

Fuente: elaboración propia

### Análisis de la varianza

Se hace 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 (escenarios) de los 5 niveles del factor con un nivel del 95,0 % de confianza. (Ver tabla 4).

Tabla 4. Resumen de varianza ANOVA

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

Fuente: elaboración propia

### Análisis de las medias

Con el fin de identificar cuáles son los escenarios diferentes se utiliza la prueba Tukey HSD. En la figura 1 se observan 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 a 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.

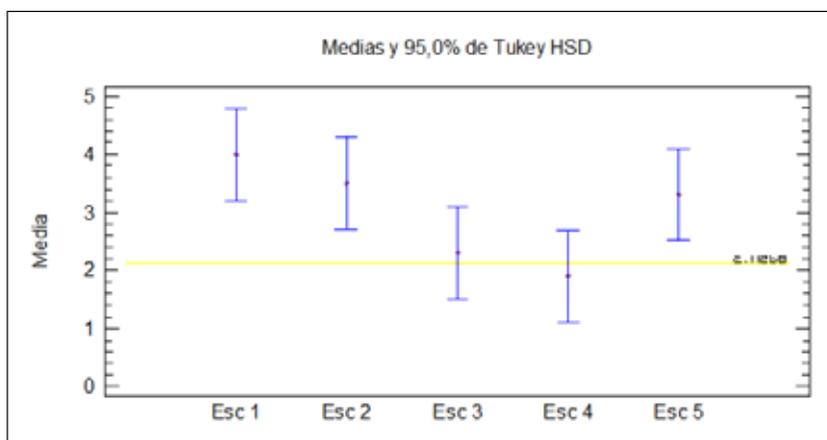


Figura 1. Estadística de medias de clasificación Vs Escenarios de Tukey HSD

Fuente: elaboración propia

Análisis de los residuos

Es importante comprobar el supuesto de normalidad. Debido a que el valor-P, tal como se muestra en la tabla 5, es 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. En la figura 2 se puede notar esta tendencia.

Tabla 5. Análisis de los residuos

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

Fuente: elaboración propia

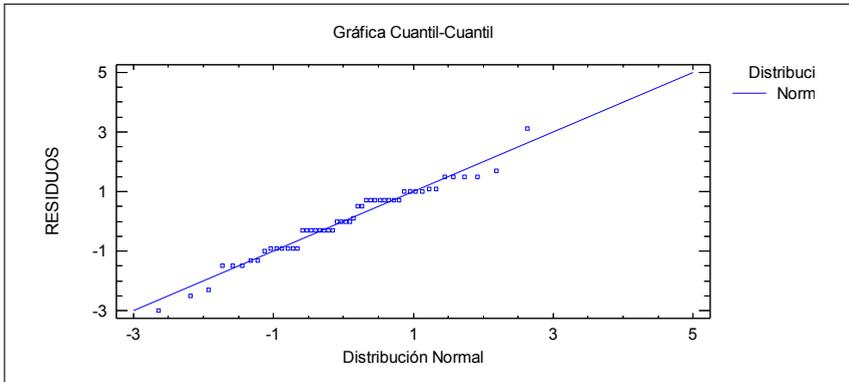


Figura 2. Probabilidad normal para residuos en clasificación

Fuente: elaboración propia

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 deben 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. (Ver figura 3).

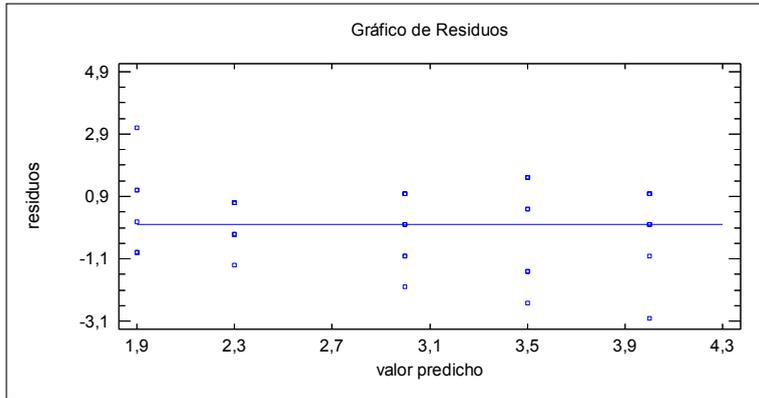


Figura 3. Residuos vs valor predicho clasificación  
Fuente: elaboración propia

En la tabla 6 se verifica este supuesto, ya que el valor-P es mayor o igual que 0,05, y no existe una diferencia estadísticamente significativa entre las desviaciones estándar, con un nivel del 95,0 % de confianza.

Tabla 6. Verificación de varianza

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

Fuente: Elaboración propia

### Independencia

La suposición de independencia en los residuos puede verificarse en la figura 4, donde se muestra un comportamiento aleatorio, por lo cual no se evidencia ningún factor que pueda asumir la falta de independencia en la medición.

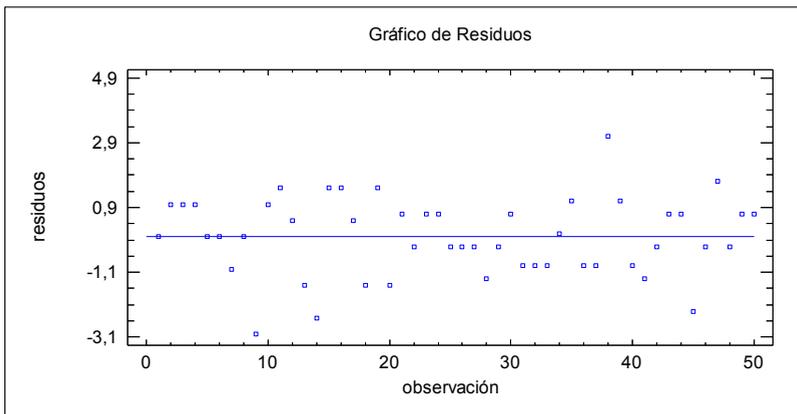


Figura 4. Independencia de residuos para clasificación  
Fuente: elaboración propia

## b. Definición del problema y factores de valoración

El factor que se estudia en este experimento es la probabilidad de ocurrencia de los escenarios en un futuro cercano. En la tabla 7 se estudia la valoración de los 10 expertos sobre los 5 escenarios propuestos de la probabilidad de ocurrencia de los escenarios.

Tabla 7. Respuesta de valoración por los expertos en los escenarios

<i>Expertos</i>	<i>Escenarios</i>				
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
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

Fuente: Elaboración propia

**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 Experto sobre los métodos de estimación.

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

**Número 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.

Donde:

$H_0 : \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = 0$  ; 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, tal como se puede apreciar en la tabla 8.

Tabla 8. Resumen estadístico para el factor de valoración

	<i>Recuento</i>	<i>Promedio</i>	<i>Desviación Estándar</i>	<i>Coficiente de Variación</i>
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 %
Total	50	3,46	1,5281	44,1649 %

Fuente: elaboración propia

### 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. (Ver tabla 9).

Tabla 9. Análisis de varianza ANOVA

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

Fuente: elaboración propia

### Análisis de medias

Al analizar la figura 5 de medias se puede observar los escenarios que hay una diferencia estadística son: Esc 1 – Esc 3, lo cual concuerda con la diferencia estadística de los demás escenarios.

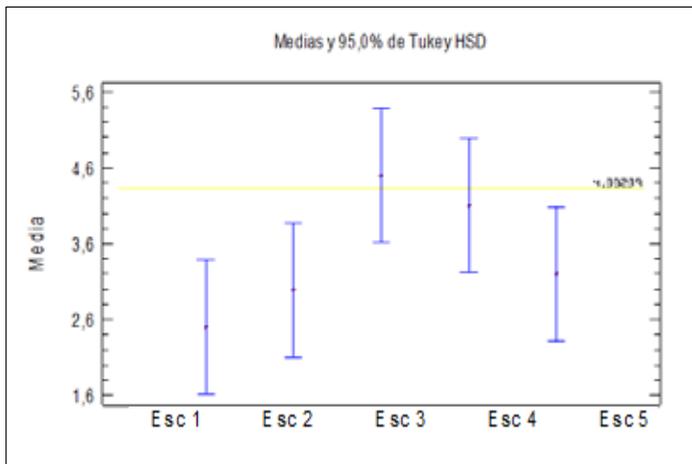


Figura 5. Estadística de medias de valoración Vs Escenarios (Tkey HSD)

Fuente: elaboración propia

### 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, (ver tabla 10). 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.

En la figura 6, se verifica lo anterior al graficar los residuos vs la distribución normal.

Tabla 10. Verificación de la varianza para la valoración

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

Fuente: elaboración propia

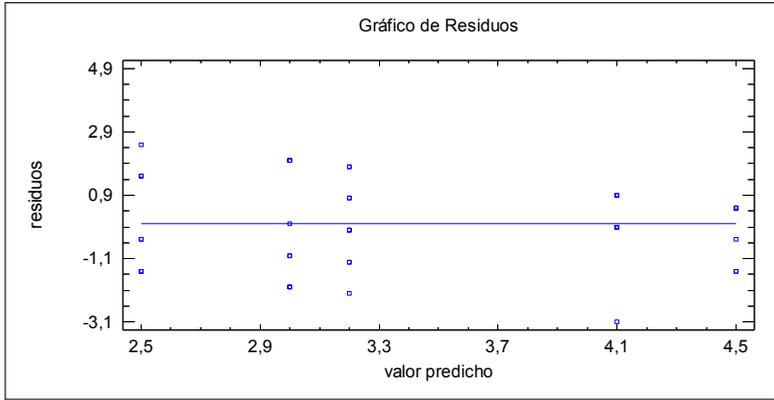


Figura 6. Residuos vs valor predicho valoración  
Fuente: elaboración propia

Independencia

La suposición de independencia en los residuos puede verificarse si se grafica el orden en que se colectó un dato contra el residuo correspondiente. Como se muestra en la figura 7, el comportamiento de los puntos es aleatorio, por lo que no hay razón para sospechar cualquier violación del supuesto de independencia.

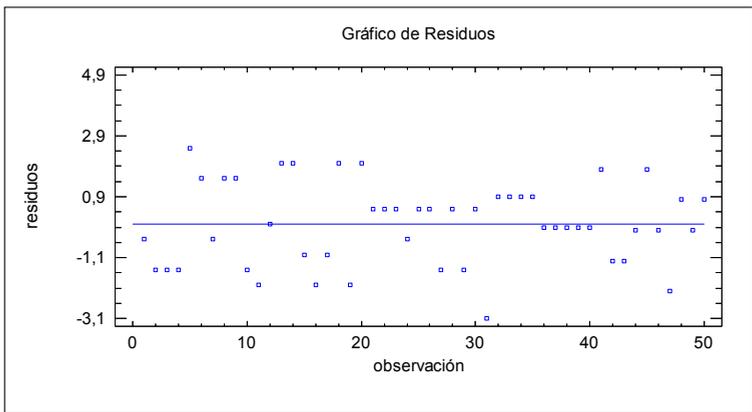


Figura 7 Independencia de valoración  
Fuente: elaboración propia

### 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 destacan las observaciones realizadas, específicamente por el experto 8 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 términos de su experiencia con el lenguaje de programación y familiaridad del proyecto. El experto 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.

## **4 MODELO PARA MEJORAR LOS PROCESOS DE ESTIMACIÓN DE COSTOS DE SOFTWARE**

El modelo planteado para mejorar los procesos de estimación de costos de software se diseña teniendo en cuenta los dos escenarios que obtuvieron mejor calificación y valoración por medio del análisis estadístico de los datos obtenidos.

Todos los factores propuestos tienen igual importancia, por lo que los agentes metodología, capacidad de equipo de trabajo y herramientas tecnológicas tienen un porcentaje del 100 %, cada uno dentro de la estructura del modelo propuesto. De acuerdo con la realimentación 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 que son la base para el modelo propuesto, donde las características y variables son consideradas con igualdad de peso.

De acuerdo con el modelo de mejora de procesos de estimación de costos de software, en la figura 8 se plantea el esquema que representa claramente la realidad de los aspectos a tener en cuenta para realizar estimaciones más cercanas a la realidad; en cada una de sus partes tiene una escala para establecer la categoría del modelo.

### Agentes de mejora de proceso de estimación de costos de software

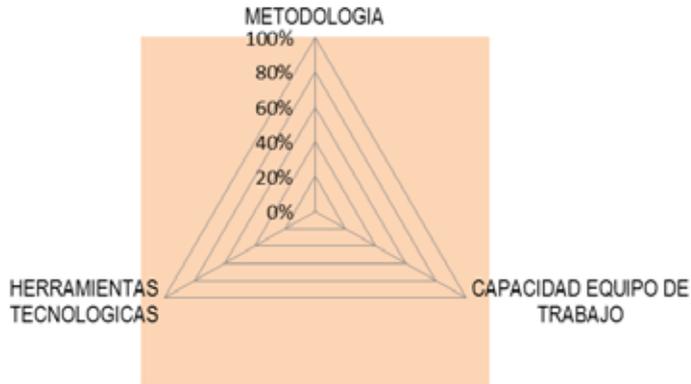


Figura 8. Agentes de mejora del proceso de estimación de costos de software  
Fuente: elaboración propia

De esta manera el modelo de análisis de mejora de procesos de estimación de costos de software consta de los siguientes agentes que definen las actividades de estimación en las empresas y cuya interacción se puede observar en la figura 9.

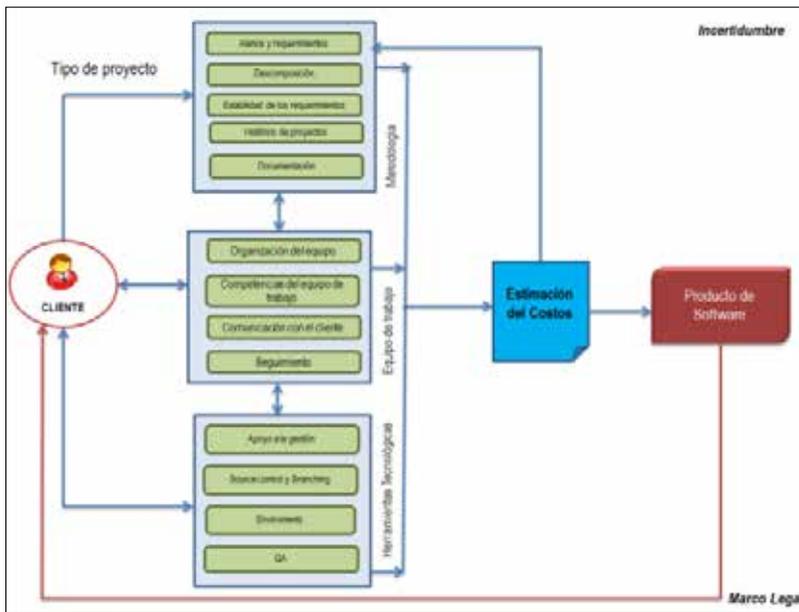


Figura 9. Modelo conceptual de mejora de procesos de estimación de costos de software para las empresas desarrolladoras.

Fuente: elaboración propia

### 4.1 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 constante y se realizan entregas parciales por medio de los llamados Sprint. En la tabla 12, se observa cada una de las categorías del agente metodología donde, dependiendo del puntaje obtenido, se podrá clasificar en alta, media o baja. Cabe resaltar que existe un conjunto de variables asociadas con este agente (ver tabla 11), en el que se especifica su grado de importancia dentro del modelo. En la tabla 17 se puede observar la ponderación para cada una de las variables del agente metodología.

Tabla 11. Categoría del agente metodología

	<i>Categoría alta</i>	<i>Categoría media</i>	<i>Categoría baja</i>
<b>AGENTE METODOLOGÍA</b>	<ul style="list-style-type: none"> <li>• Se realizan reuniones con el cliente y futuros usuarios donde se establecen los requisitos funcionales y no funcionales del sistema</li> <li>• Descomposición de requerimientos en actividades y tareas específica.</li> <li>• Asignación de prioridad por parte del cliente a los requerimientos.</li> <li>• Requerimientos estables a lo largo de los sprint</li> <li>• Manejo de base de datos histórica y personal con alta experiencia en proyectos similares.</li> <li>• Documentación de especificación de requisito y del diseño del software que soportan las actividades del equipo.</li> <li>• En cada sprint va creciendo el manual del usuario.</li> </ul>	<ul style="list-style-type: none"> <li>• Se establecen con el cliente los requisitos, pero tienen a ser poco claros al existir problemas en la comunicación.</li> <li>• Descomposición de requerimientos priorizados por nivel de complejidad.</li> <li>• Personal con experiencia empírica en proyectos similares.</li> <li>• Volatilidad de los requisitos al no tener claro el cliente algunas funcionalidades del sistema</li> <li>• Documentación poco clara para el cliente.</li> </ul>	<ul style="list-style-type: none"> <li>• Se establecen con el cliente los requisitos, pero tienen a ser poco claros al existir problemas en la comunicación.</li> <li>• se descomponen los requisitos en actividades o tareas, pero se presentan retrasos al momento de realizar el desarrollo ya que son poco claros o por inconvenientes técnicos.</li> <li>• Volatilidad de los requisitos al no tener claro el cliente algunas funcionalidades del sistema</li> <li>• Personal sin experiencia en proyectos similares.</li> <li>• Ausencia de base de datos histórica</li> <li>• Manejo de documentación poco clara de los requerimientos.</li> <li>• El manual de usuario y sistema se genera al final del proyecto</li> </ul>

Fuente: elaboración propia

Tabla 12. Variables del agente metodología

<i>Variables del agente Metodología</i>				
<i>Alcance y requerimientos</i>	<i>Descomposición</i>	<i>Estabilidad de los requerimientos</i>	<i>Histórico de proyectos</i>	<i>Documentación</i>
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
20 %	20 %	20 %	20 %	20 %
Porcentaje total: 100 %				

Fuente: elaboración propia

## 4.2 Capacidad de equipo de trabajo

La capacidad del equipo de trabajo es el insumo más importante, pues se refiere a la relación del número de personas trabajando para un proyecto, ya que partiendo del rol que desempeña cada uno para su ejecución, las habilidades y su experiencia adquirida, el esfuerzo va a ser mucho menor para la ejecución del proyecto. En la tabla 13 se observa cada una de las categorías del agente equipo de trabajo, donde, dependiendo del puntaje obtenido, se podrá clasificar en categoría alta, media o baja. Cabe resaltar que existen un conjunto de variables asociadas a este agente (ver tabla 14) en el que se especifica su grado de importancia dentro del modelo. En la tabla 18 se puede observar la ponderación para cada una de las variables del agente equipo de trabajo.

Tabla 13. Categoría del agente equipo de trabajo

	<i>Categoría alta</i>	<i>Categoría media</i>	<i>Categoría baja</i>
<b>AGENTE EQUIPO DE TRABAJO</b>	<ul style="list-style-type: none"> <li>La empresa cuenta con un equipo sólido con roles establecidos para cada fase del ciclo de vida del proyecto.</li> <li>El personal se encuentra capacitado para utilizar las diferentes herramientas de desarrollo</li> <li>Políticas de capacitación para elevar el nivel de competencias del equipo</li> <li>Existencia de personal altamente capacitado</li> </ul>	<ul style="list-style-type: none"> <li>La empresa cuenta con un equipo de trabajo, pero falta mayor organización en sus funciones y en los roles que ejercen a lo largo del ciclo de vida del proyecto</li> <li>Interés de la empresa y del equipo por mantenerse actualizado por medio de capacitaciones</li> <li>Barreras en la comunicación con el cliente para res-</li> </ul>	<ul style="list-style-type: none"> <li>Se asumen roles que no corresponden a algunos integrantes del equipo al no estar bien definido</li> <li>No se tienen planes de capacitación para los empleados</li> <li>El equipo de trabajo ha adquirido los conocimientos empíricos en las funciones de cada cargo</li> </ul>

	<ul style="list-style-type: none"> <li>• Constante y efectiva retroalimentación por parte del cliente del progreso del proyecto</li> <li>• El líder del proyecto realiza reuniones con el equipo de trabajo con el fin de evaluar el cronograma de actividades</li> <li>• Manejo de indicadores de cumplimiento</li> </ul>	<p>ponder cuestionamientos funcionales</p> <ul style="list-style-type: none"> <li>• Alto tiempo de respuesta a las inquietudes por parte del equipo</li> <li>• Barrera del idioma para establecer una comunicación efectiva con el cliente</li> <li>• Reuniones del líder de proyecto con el equipo poco efectivas</li> </ul>	<ul style="list-style-type: none"> <li>• Poca retroalimentación a inquietudes del equipo por parte del cliente</li> <li>• No se cumple un cronograma regular de reuniones con el equipo de trabajo para hacer seguimiento a las tareas y requerimientos.</li> </ul>
--	--	---	---

Fuente: elaboración propia

Tabla 14. Variables del agente equipo de trabajo

<i>Variables del agente equipo de trabajo</i>			
<i>Organización del equipo</i>	<i>Competencias del equipo del trabajo</i>	<i>Comunicación con el cliente</i>	<i>Seguimiento.</i>
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.
25 %	25 %	25 %	25 %
Porcentaje total: 100 %			

Fuente: elaboración propia

### 4.3 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. Si la empresa desarrolladora posee un producto que comercializa o con la cual tiene experiencia, puede decidir qué tipo de herramienta utilizar. En la tabla 16, se observa cada una de las categorías del agente herramientas tecnológicas donde dependiendo del puntaje obtenido se podrá clasificar en categoría alta, media o baja. Cabe resaltar que existe un conjunto de variables asociadas a este agente (ver tabla 15) en el que se especifica su grado de importancia dentro del modelo. En la tabla 19, se puede observar la ponderación para cada una de las variables del agente herramientas tecnológicas.

Tabla 15. Categoría del agente herramientas tecnológicas

	<i>Categoría alta</i>	<i>Categoría media</i>	<i>Categoría baja</i>
<b>AGENTE HERRAMIENTAS TECNOLÓGICAS</b>	<ul style="list-style-type: none"> <li>• Manejo de gestión de proyectos para apoyar la medición de avances del cronograma de trabajo, establecer prioridad de los requerimientos.</li> <li>• El cliente tiene acceso a la herramienta de gestión de proyecto.</li> <li>• Manejo de herramientas para controlar el código que permite revertir errores locales.</li> <li>• El equipo de desarrollo utiliza diferentes environments para aislar los bugs y mantener el producto estable</li> <li>• Manejo de metodologías y métodos para asegurar la calidad del producto</li> </ul>	<ul style="list-style-type: none"> <li>• Manejo de gestión de proyectos para apoyar la medición de avances del cronograma de trabajo, establecer prioridad de los requerimientos.</li> <li>• Manejo de herramientas para controlar el código que permite revertir errores locales.</li> <li>• El equipo de desarrollo utiliza diferentes environments para aislar los bugs y mantener el producto estable.</li> <li>• El equipo de desarrollo tiene un environment de pruebas pero no usa métodos de QA desde el segundo cero.</li> </ul>	<ul style="list-style-type: none"> <li>• Se asumen roles que no corresponden a algunos integrantes del equipo al no estar bien definido.</li> <li>• El equipo de desarrollo no maneja una herramienta formal de gestión del proyecto.</li> <li>• 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.</li> <li>• El equipo usa un solo environment para el desarrollo, pruebas y el release del producto a producción.</li> <li>• El equipo no tiene environment de pruebas ni tecnologías para QA definidas.</li> </ul>

Fuente: elaboración propia

Tabla 16. Variables del agente herramientas tecnológicas

<i>Variables del agente Herramientas Tecnológicas</i>			
<i>Apoyo a la gestión</i>	<i>Source control y Branching</i>	<i>Enviroments</i>	<i>QA</i>
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 %
Porcentaje total: 100 %			

Fuente: elaboración propia

Tabla 17. Ponderación para variables agente metodología

<i>Variable</i>	<i>Escala de la variable</i>	<i>Puntos</i>
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

Fuente: elaboración propia

Tabla 18. Ponderación para variables agente Equipo de trabajo

<i>Variable</i>	<i>Escala de la variable</i>	<i>Puntos</i>
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

Fuente: elaboración propia

Tabla 19. Ponderación para variables agente herramientas tecnológicas

<i>Variable</i>	<i>Escala de la variable</i>	<i>Puntos</i>
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 requerimientos, 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

Fuente: elaboración propia

## 5 CONCLUSIONES

El modelo propuesto se basó en un conjunto de escenarios o posibles ambientes de cómo las empresas realizarían sus procesos de estimaciones de costos de software. 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 llevó a cabo un proceso de clasificación y validación utilizando el método Delphi.

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

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 complicado 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.

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 evidenció que los métodos no se aplicaban en su totalidad, sino que se adaptaron a un contexto muy particular de acuerdo con las condiciones socio económicas de la región caribe colombiana. Es por esto que, para la segunda ronda, 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, sin embargo, resultados similares.

## REFERENCIAS

- [1] J. A. I. P. Pow-Sang Portillo, Ricardo, “Estimación y Planificación de Proyectos Software con Ciclo de Vida Iterativo-Incremental y empleo de Casos de Uso,” p. 6, 2005.
- [2] S. McConnell, *Software Estimation: Demystifying the Black Art*: Microsoft Press A Division of Microsoft Corporation, 2006.

- [3] Azam, F., Qadri, S., Ahmad, S., Khan, K., Siddique, A. B., & Ehsan, B. (2014). Framework Of Software Cost Estimation By Using Object Orientated Design Approach. *IJSTR*, 3(8), 97-100.
- [4] CHAOS, E. (2004). Extreme Chaos 2004–3rd quarter research report. The Standish Group International.
- [5] H. Al-Sakran, “Software Cost Estimation Model Based on Integration of Multi-agent and Case-Based Reasoning,” *Journal of Computer Science* vol. 2, pp. 276-282, 2006.
- [6] V. Bozhikova and M. Stoeva, “An Approach for Software Cost Estimation,” presented at the International Conference on Computer Systems and Technologies, 2010.
- [7] A. Magazinius and R. Feldt, “Exploring the Human and Organizational Aspects of Software Cost Estimation,” *ACM SIGSOFT Software Engineering Notes*, 2010.
- [8] Pendharkar, P. C., Subramanian, G. H., & Rodger, J. (2005). A probabilistic model for predicting software development effort. *Software Engineering, IEEE Transactions on*, 31(7), 615-624.
- [9] Shepperd, M. (2014). Cost Prediction and Software Project Management. In *Software Project Management in a Changing World* (pp. 51-71). Springer Berlin Heidelberg.
- [10] N. Kavantzias, D. Burdett, G. Ritzinger, T. Fletcher, and Y. Lafon, “Web Services Choreography Description Language (WS-CDL) version 1,” 2004.
- [11] Xu, G., & Gutiérrez, J. A. (2007). An Exploratory Study of “Killer Applications” and Critical Success Factors in M-Commerce. *Web Technologies for Commerce and Services Online*, 231.
- [12] Skulmoski, G., Hartman, F., & Krahn, J. (2007). The Delphi method for graduate research. *Journal of Information Technology Education: Research*, 6(1), 1-21.