**Estimación de Costos**

La estimación de costos y esfuerzos sigue siendo una de las tareas más difíciles en la gestión de un proyecto de software. En la actualidad existen técnicas que permiten realizar esta labor, aunque, lamentablemente, aún no hay técnicas maduras específicas para enfoques de desarrollo como la orientación a objetos o como los sistemas muy específicos. A este problema se suma la escasa información proporcionada por las técnicas de estimación existentes para su aplicación a ciclos de vida de desarrollo de software diferente al de cascada, como, por ejemplo, los ciclos de vida iterativo-incrementales o en espiral [1].

Estas técnicas pueden ser:

* Técnicas descendentes (de arriba hacia abajo): Las técnicas descendentes incluyen información histórica, analogía y proporción. En general las estimaciones que confían más en la división del trabajo detallada son conocidas como ascendentes (o de abajo hacia arriba).
* Técnicas descendentes (de abajo hacia arriba): Son típicamente más rápidas y fáciles de integrar ya que estamos estimando a nivel de todo el proyecto. También resultan menos precisas, con excepción de las estimaciones que se basan en información histórica o bien en la analogía de proyectos similares.

Hay que tener en cuenta que algunos de los principales factores que afectan el costo del software, lo cuales son principalmente:

* La capacidad del equipo o programador(es).
* La complejidad del producto.
* El tamaño del proyecto.
* El tiempo disponible para ejecutar el proyecto.
* La calidad que se requiere.
* Nivel tecnológico (normalmente viene dado por el lenguaje a utilizar).

Todos estos factores pueden hacer que se aumente o disminuya el margen de la estimación teniendo en cuenta los riesgos que se quieran o puedan asumir.

**Delphi**

El método Delphi por medio de un conjunto de rondas con un panel de expertos identifica 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. 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 [2].

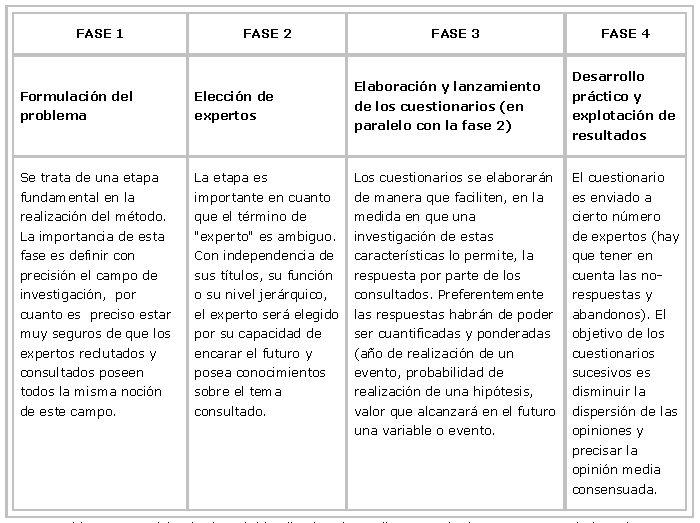


Tabla 1. Fases del método Delphi aplicadas al estudio. Tomado de Kavantzas et al. (2004), Skulmoski et al. (2000), Xu y Gutiérrez (2006)

La lista de escenarios a evaluar por los expertos proviene de la revisión de la literatura sobre el tema. Se presentaron las categorías de Juicio Experto, Estimación por analogía, Precio para ganar y estimaciones por medio del uso de aplicaciones sistematizadas.

A continuación, se detalla cada uno de los escenarios propuestos que fueron evaluados por los expertos:

| **Escenario A**  **Juicio experto** | **Escenario B**  **Analogía** | **Escenario C**  **precio para ganar** | **Escenario D**  **Aplicaciones sistematizadas** |
| --- | --- | --- | --- |
| 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. | 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. | La empresa desarrolladora de software considera que el costo del proyecto del software está en función de lo que el cliente está dispuesto a paga 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. | 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. |

Tabla 2. Evaluación de Escenarios por expertos. Ronda 1. [2]

| **Expertos** | **Escenarios** | | | | |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **OBSERVACIÓN** |
| 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 presupuesto 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 la misma empresa. | 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 |  |

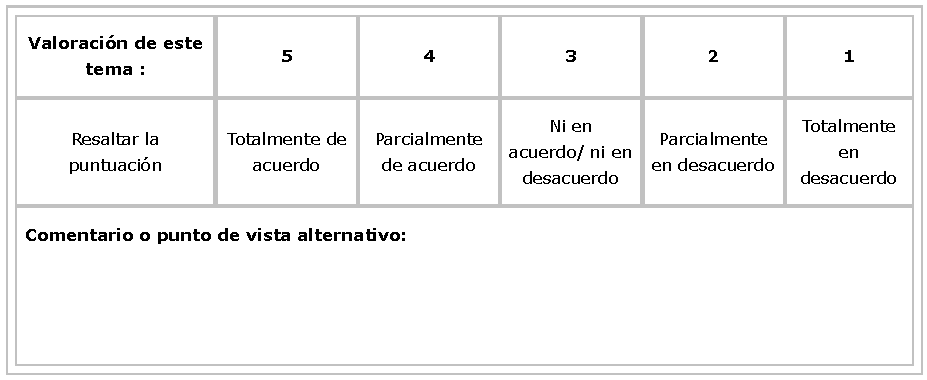
**Segunda ronda**

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.

| Escenario A1 | Escenario B1 | Escenario C2 | Escenario D1 | Escenario E1 |
| --- | --- | --- | --- | --- |
| Una de las formas de realizar la estimación de esfuerzo o tamaño relativo de las tareas de desarrollo e basa en utilizar una baraja de cartas que se encuentran enumeradas mostrando la secuencia Fibonacci: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100. Se utiliza esta secuencia para identificar la incertidumbre de cada una de las estimaciones. | 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. | 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. | 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 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. |



**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.

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.

**Puntos de Función**

Consiste en asignar una cantidad de “puntos” a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos. Siempre tratando de considerarlo desde el punto de vista del usuario. Una vez teniendo este recuento, teniendo en cuenta 14 características generales del sistema (que tienen que ver con la arquitectura de la aplicación, sus requisitos de carga y rendimiento, complejidad de cálculos, etc.) valorados de 0 a 5 en función de su grado de influencia, se calcula un factor de ajuste al recuento. Aplicando este factor de ajuste, se obtiene el recuento final. [1]

Traducir en un número el tamaño de la funcionalidad que brinda un producto de software.

Asignarle un valor numérico a la funcionalidad, respecto a la complejidad.

* Desde el punto de vista del usuario
* Suma ponderada de características del producto

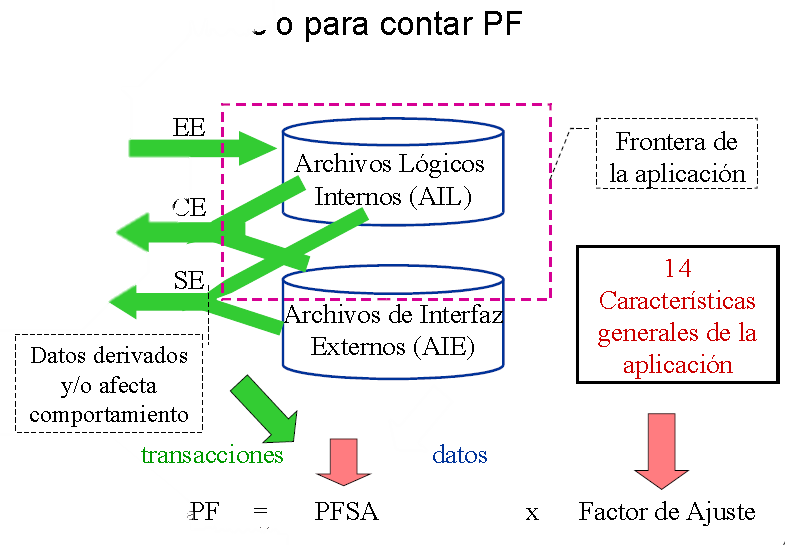
**Transacciones:**

* Número de Entradas Externas (EE)
* Número de Salidas Externas (SE)
* Número de Consultas Externas (CE)

**Datos:**

* Número de Archivos Interfaz Lógicos (AIL)
* Número de Archivos Interfaz Externa (AIE)

**Modelo para contar PF**



**Ficheros de Salida**



**Clasificación de los Ficheros Lógicos o Internos**



**Clasificación de los Ficheros de Interfaz**

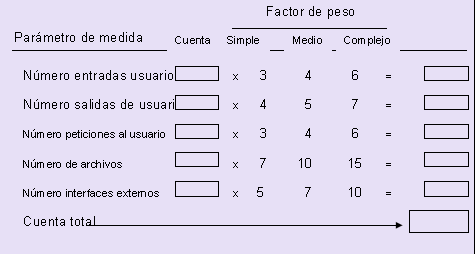


**Factores de Ajuste de Complejidad:**

Evaluar cada factor de 0 a 5

1. Sin influencia
2. Incidental
3. Moderado
4. Medio
5. Significativo
6. Esencial

Factor de Peso



PF = Cuenta Total x [0,65 + 0,01 x SUM(Fi)]

Fi: valores de ajuste de complejidad

**EJEMPLO:**

Desarrollo de un proyecto matemático de métodos numéricos [3]:

Datos:

Productividad = 1000 pf / per-mes

Sueldo = 3000 pesos / per-mes



* Copia de seguridad y recuperación 2
* Comunicaciones 3
* Proceso distribuido 1
* Rendimiento crítico 4
* Entorno operativo existente 4
* Entrada de datos online 5
* Transacciones entradas en varias pantallas 1
* Archivos maestros actualizados online 2
* Complejidad valores dominio información 3
* Complejidad procesamiento interno 3
* Código diseñado para reutilización 4
* Conversión en diseño 1
* Instalaciones múltiples 3
* Aplicación diseñada para cambios 5

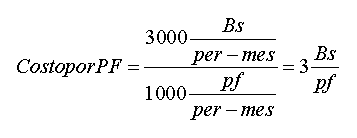
**Calculo puntos Función**

PF = Cuenta Total x [0,65 + 0,01 x SUM(Fi)]

PF = (366) x [0,65 + 0,01x (2+3+1+4+4+5+1+2+3+3+4+1+3+5)]

PF= (366) x (1.06) = 387,96.

**Calcular el costo estimado del proyecto:**



Costo del proyecto = PF \* Costo por PF = 387,96 pf \* 3Bs / pf

Costo del proyecto = 1163,88 Bs.

**Técnica LDC**

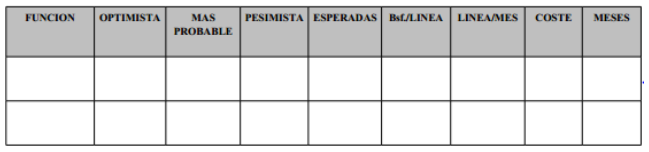
Línea de código: Es una medida propuesta inicialmente cuando los programas se escribían en tarjetas, con una línea por tarjeta. Actualmente los lenguajes permiten escribir varias sentencias en una línea, o una misma sentencia en varias líneas.

Las LDC miden en forma directa el tamaño del producto de software. Se calculan contando las instrucciones de código fuente de cada elemento del producto de software excluyendo, generalmente, los comentarios [4].

Antes de adoptar esta métrica, la organización debe definirla en forma exhaustiva. Esta definición debe respetarse, ya que podría atentar a la integridad de los datos del proyecto. Cuando se utiliza LDC como variable de estimación, la descomposición funcional es absolutamente esencial, a menudo se lleva hasta considerables niveles de detalle.

Usando PF es la variable de estimación es menos detallado. También, debe de tenerse en cuenta que mientras que LDC se estima directamente, PF se determina indirectamente mediante la estimación del número de entradas, salidas, archivos de datos y peticiones externas, entre otras. Entonces, se calcula el valor esperado de LDC. El valor esperado para la variable de estimación, E, se obtiene como una medida ponderada de las estimaciones LDC óptima (a), más probable (m) y pesimista (b). Esta técnica trata de definir el tiempo y el costo del proyecto en base a la cantidad de líneas de código se tienen que escribir, cual es el costo por línea y cuantas líneas de código desarrollamos en un mes [4].

Cuadro para calcular los LDC:



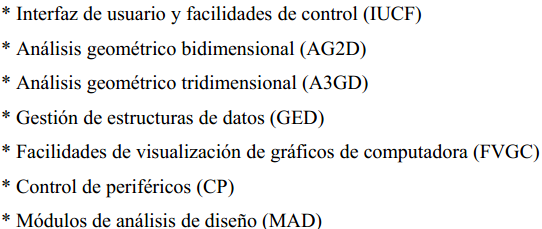
**Pasos para el cálculo LDC:**

* Descomponer el problema
* Estimar valores para columnas de líneas de código a escribir
* Calcular columna esperada

Ejemplo de utilización LDC

Considerar un paquete de software a desarrollar para una aplicación de diseño asistido por un computador. Revisando la especificación del sistema encontramos que el software va a ejecutarse en una estación de trabajo de microcomputadora y se conectara con varios periféricos gráficos incluyendo ratón, digitalizador, pantalla en color de alta resolución y una impresora de alta resolución.

La evaluación del alcance indica que se requiere las siguientes funciones principales:



**Solución**



# **Referencias**

| [1] | J. I. Acedo, «http://programacion.jias.es,» 01 Marzo 2017. [En línea]. Available: http://programacion.jias.es/2017/03/tecnicas-de-estimacion-para-el-esfuerzo-en-proyectos/. [Último acceso: 01 Octubre 2018]. |
| --- | --- |
| [2] | E. Puerta y H. Pérez, «Aplicación del método Delphi para establecer un modelo conceptual de estimación de costos de software,» *Espacios,* vol. 38, nº 36, p. 11, 2017. |
| [3] | A. Pastrana, «slideshare.net,» 16 Junio 2010. [En línea]. Available: https://es.slideshare.net/antoniopastrana/tecnicas-de-estimacion-de-costos-de-proyecto-software. [Último acceso: 01 Octubre 2018]. |
| [4] | C. C. I. y. E. M. Cárdenas., «http://arfduoc.blogspot.com,» [En línea]. Available: http://arfduoc.blogspot.com/p/estimaciones-ldc-y-pf.html. [Último acceso: 01 Octubre 2018]. |