



INGENIERÍA DE SOFTWARE III

UNIDAD IV – ESTIMACIÓN DE PROYECTOS
SOFTWARE

Prof. MS. Luis Gilberto Salinas – Prof. Ing. Julia Talavera

CONTENIDO

- ❑ Introducción al Proceso de Estimación
- ❑ Estimación de los Puntos de Función
- ❑ Estimación de Primer Orden de Jones
- ❑ Estimación del Esfuerzo – COCOMO II

Introducción al proceso de estimación

¿Qué es una estimación de software?

Una estimación de software es una predicción de cuánto tiempo durará o costará su desarrollo y mantenimiento.

Si se trata de una estimación de tiempo, el esfuerzo puede expresarse en horas-hombre, si se trata de estimación de costo, se puede expresar de manera monetaria.

El reto de elaborar estimaciones de software, es realizar predicciones realistas, basándose en información incompleta e incierta.

Introducción al proceso de estimación

En Ingeniería de software y gestión de proyectos de software, las estimaciones se utilizan para:

- ☐ Desarrollar planes de proyectos.
- ☐ Elaborar planificaciones de iteración en desarrollo de software.
- ☐ Elaborar presupuestos
- ☐ Realizar análisis de inversión.
- ☐ Fijación de precios de un software para un cliente empresarial.
- ☐ Análisis para determinar el precio en software dirigido al consumidor.
- ☐ Para planificar estrategia cuando se dispone a participar en subastas de contratos en los que participan varios proveedores.

Introducción al proceso de estimación

ALGUNAS ESTIMACIONES SON REALIZADAS CUIDADOSAMENTE y otras se hacen a ojo. La mayoría de los proyectos rebasan los límites de sus planificaciones estimadas entre el 25 y el 100 por 100, pero muy pocas organizaciones han logrado realizar una predicción de la planificación con una precisión de un 10 por 100, e incluso se ha llegado en algún caso a un 5 por 100 (Jones, 1994).

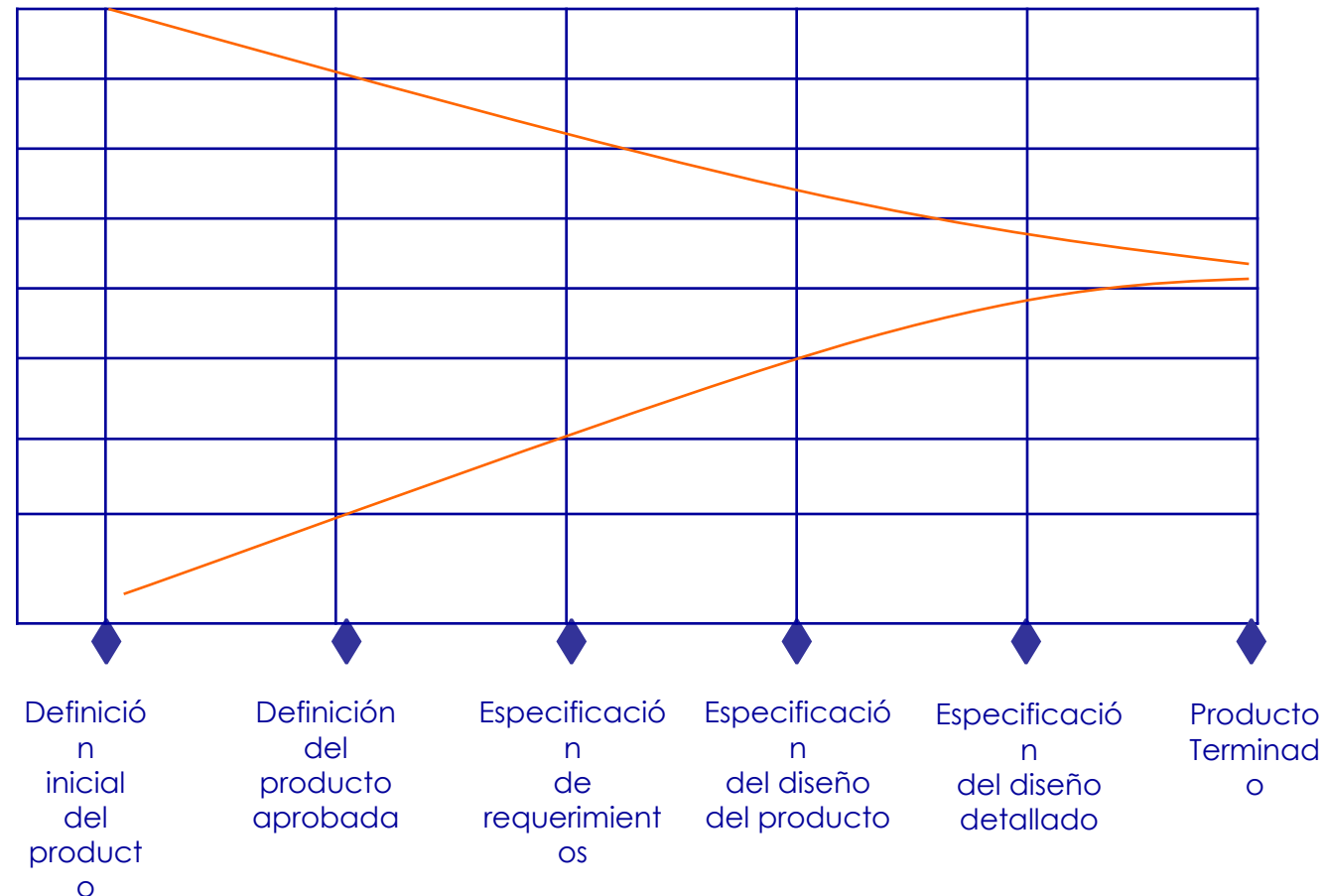
Introducción al proceso de estimación

Los investigadores han descubierto que las estimaciones de proyectos entran dentro de precisiones previsibles en varios estados del proyecto. El gráfico de convergencia de estimación muestra como la estimación se hace más precisa conforme avanza el proyecto.

Introducción al proceso de estimación

**Costo del proyecto
(esfuerzo y
tamaño)**

**Duración del
proyecto**

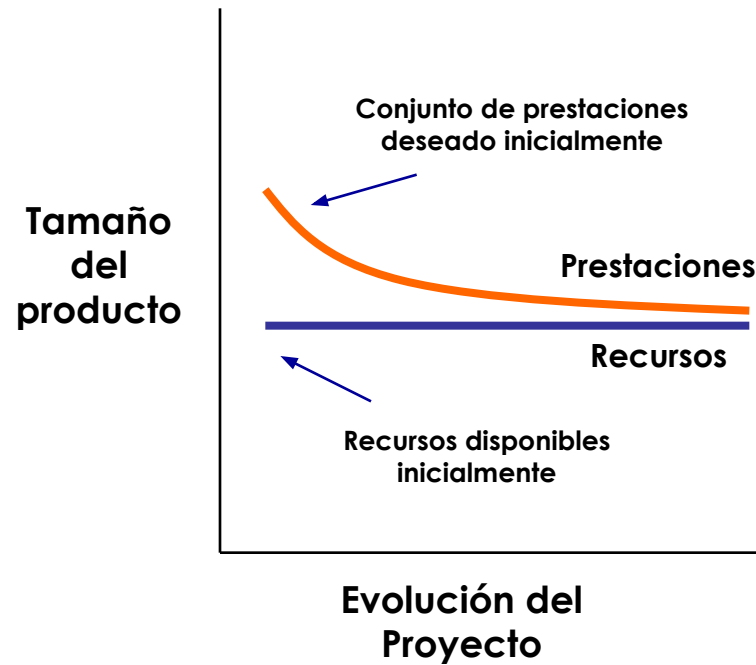


Introducción al proceso de estimación

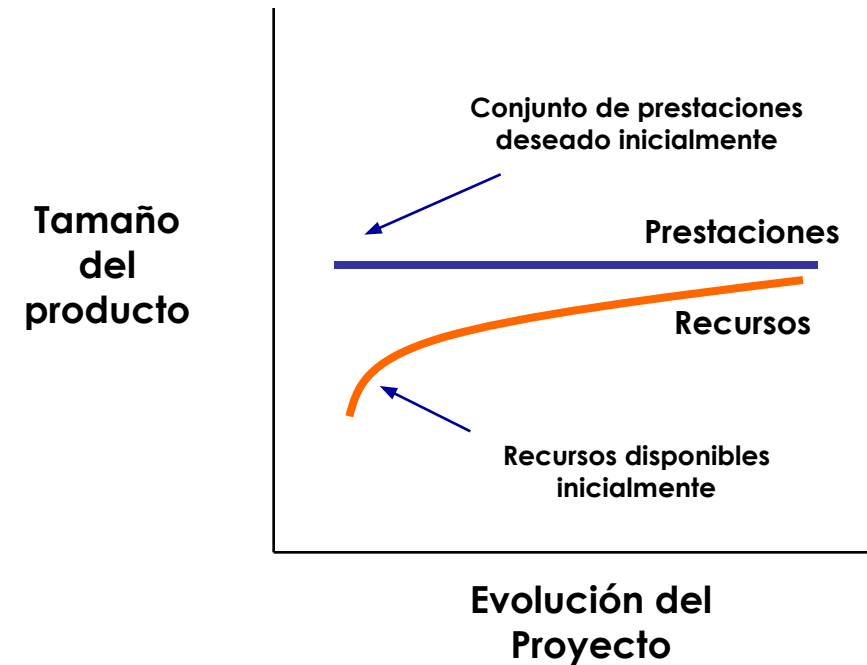
La mayoría de los clientes de software desean inicialmente más de lo que se pueden permitir, tienen que adaptar sus ideas sobre el producto a los recursos con los que están dispuestos a comprometerse. A veces el cliente deseará adaptar tanto los recursos como el conjunto de prestaciones para llegar a un acuerdo entre ambos.

Introducción al proceso de estimación

Las prestaciones se adaptan para adecuarse a los recursos disponibles



Los recursos se adaptan para adecuarse a las prestaciones deseadas

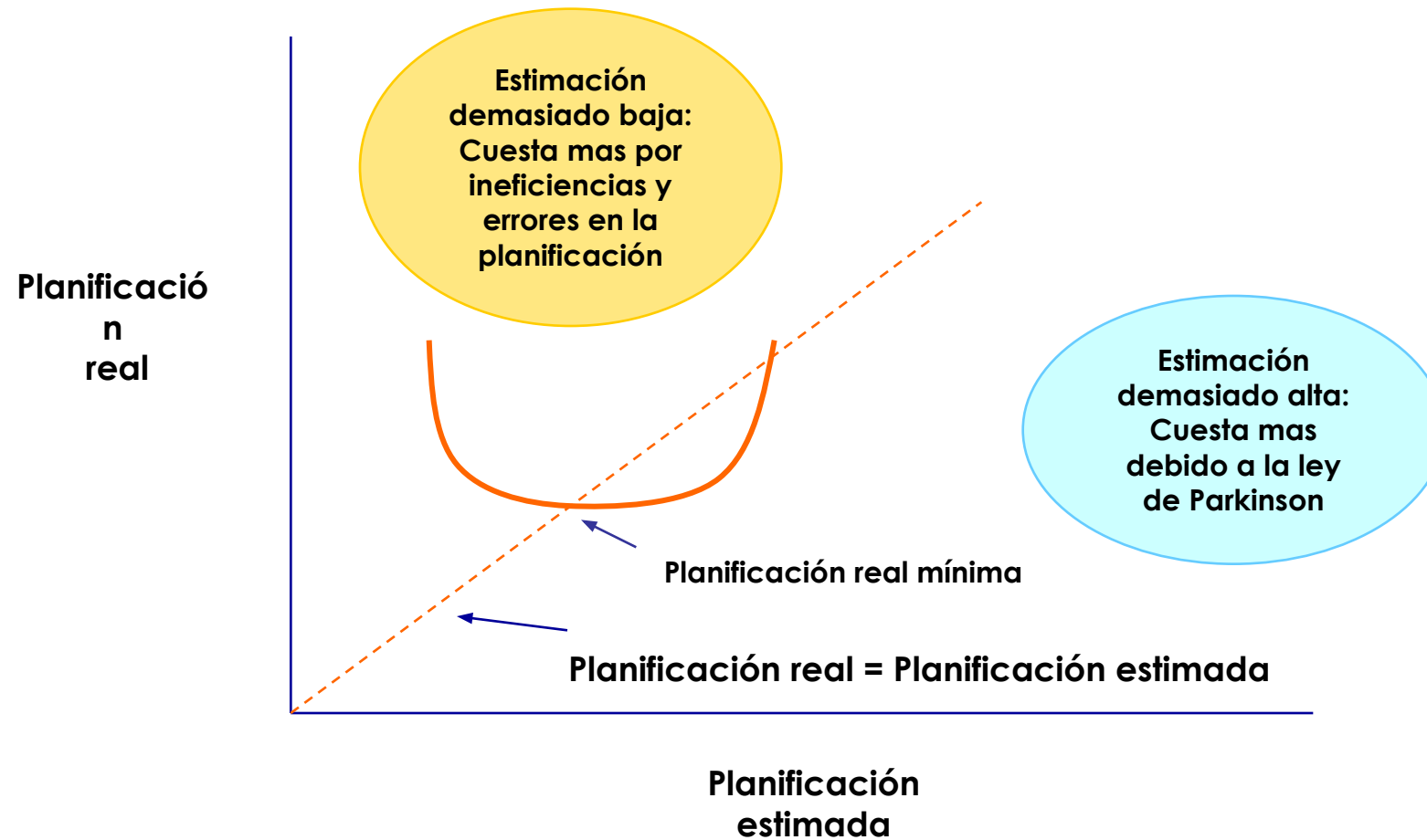


Introducción al proceso de estimación

Si los clientes desean la planificación más corta posible, no deberían presionar para reducir la estimación o proporcionar unas estimaciones “precisas” equivocadas.

La planificación real más corta resulta de la programación planificada más precisa. Si la estimación es demasiado baja, la planificación ineficiente conducirá al costo real del proyecto. Si la estimación es demasiado alta, según la ley de Parkinson (el trabajo se reparte para cubrir el tiempo disponible), conducirá al costo real del proyecto.

Introducción al proceso de estimación



Introducción al proceso de estimación

El proceso para crear una buena planificación, consta de tres pasos:

- ❑ **Estimar el tamaño del producto (número de líneas de código o puntos de función).** Algunos proyectos saltan directamente a la estimación de la planificación, pero una estimación efectiva necesita estimar primero el tamaño del software para poder construirlo. Este paso es con frecuencia el más difícil intelectualmente, y ésta podría ser una de las razones por las que se lo suele saltar.
- ❑ **Estimar el esfuerzo (personas-mes).** Si tiene una estimación exacta del tamaño y los datos previos de la organización en proyectos similares, es fácil realizar la estimación del esfuerzo.
- ❑ **Estimar la planificación (meses).** Una vez que se han estimado el tamaño y el esfuerzo, estimar la planificación resulta casi trivial. Pero obtener la aceptación de una estimación de la planificación realista puede ser la parte más difícil del proyecto.

Introducción al proceso de estimación

Se puede estimar el tamaño de un proyecto de varias formas:

- ❑ **Utilizar un enfoque algorítmico**, como los puntos de función o casos de uso, para estimar el tamaño del programa a partir de las prestaciones.
- ❑ **Utilizar un software de estimación** para estimar el tamaño del programa, a partir de la descripción de las prestaciones del programa (pantallas, diálogos, archivos, tablas de la base de datos, etc.).
- ❑ **Si ya se ha trabajado en proyectos similares**, y se conoce el tamaño, estimar cada una de las partes principales del nuevo sistema como un porcentaje del tamaño de una parte similar del sistema antiguo. Estimar el tamaño total del sistema nuevo sumando los tamaños estimados de cada una de las partes.

Estimación de los puntos de función

Un punto de función es una medida sintética del tamaño del programa, que se suele utilizar en los primeros estados del proyecto (Albrecht, 1979). Los puntos de función son más fáciles de determinar a partir de la especificación de los requerimientos que las líneas de código, y proporcionan una medida más exacta sobre el tamaño del programa. Existen diferentes métodos para contar los puntos de función; el que describimos se basa en "1984 IBM Method", que es la base del método actual de IBM y del International Function Point User Groups's (IFPUG's).

Estimación de los puntos de función

El número de puntos de función en un programa se basa en el número y la complejidad de cada uno de los elementos siguientes:

- ❑ Entradas. Pantallas, formularios, cuadros de diálogo, controles o mensajes, a través de los cuales un usuario final o cualquier otro programa pueda añadir, borrar o cambiar datos del programa.
- ❑ Salidas. Pantallas, informes, gráficos o mensajes que el programa genera para el usuario final o cualquier otro programa.
- ❑ Consultas. Combinaciones de entrada/salida, en las que cada entrada genera una salida simple e inmediata. El término surge en el mundo de las bases de datos para hacer referencia a una búsqueda directa de un dato específico, usando generalmente una única clave. En las aplicaciones GUI modernas, la separación entre las consultas y las salidas es confusa; sin embargo, generalmente las consultas recuperan datos directamente de una base de datos y muestran sólo el formato elemental, mientras que las salidas pueden procesar, combinar o resumir datos complejos y presentar muchos formatos.

Estimación de los puntos de función

- ❑ **Archivos lógicos internos.** Los principales grupos lógicos de datos de usuarios finales o información de control que están completamente controlados por el programa. Un archivo lógico podría constar de un único archivo plano o de una sola tabla en una base de datos relacional.
- ❑ **Archivos de interfaz externos.** Archivos controlados por otros programas, con los que el programa va a interactuar. Esto incluye cada uno de los principales grupos de datos lógicos o información de, control que entre o salga del programa. La terminología que se utiliza en el enfoque de los puntos de función está poco orientada a las bases de datos. El enfoque básico funciona bien, para toda clase de software, pero si no se están construyendo sistemas intensivos de bases de datos tendrá que ajustarse al propio entorno.

Estimación de los puntos de función

Características del Sistema	Complejidad Baja	Complejidad Media	Complejidad Alta
Número de entradas	X 3	X 4	X 6
Número de salidas	X 4	X 5	X 7
Consultas	X 3	X 4	X 6
Archivos lógicos internos	X 7	X 10	X 15
Archivos de interfaz externos	X 5	X 7	X 10

Estimación de los puntos de función

Características del Sistema	Complejidad Baja	Complejidad Media	Complejidad Alta
Número de entradas	$6 \times 3 = 18$	$2 \times 4 = 8$	$3 \times 6 = 18$
Número de salidas	$6 \times 4 = 28$	$7 \times 5 = 35$	$0 \times 7 = 0$
Consultas	$0 \times 3 = 0$	$2 \times 4 = 8$	$4 \times 6 = 24$
Archivos lógicos internos	$5 \times 7 = 35$	$2 \times 10 = 20$	$3 \times 15 = 45$
Archivos de interfaz externos	$9 \times 5 = 45$	$0 \times 7 = 0$	$2 \times 10 = 20$
Total de puntos de función sin ajustar			304
Multiplicador			0,95
Total de puntos de función ajustados			289

Estimación de los puntos de función

Valores de ajuste de complejidad

Descripción	Valor
Sin influencia	0
Accidental, Influencia no significativa	1
Moderado, Influencia moderada	2
Medio, Influencia media	3
Significativo, Influencia significativa.	4
Esencial, Influencia esencial	5

Estimación de los puntos de función

Nro.	Preguntas	Vi
1	Requiere el sistema copias de seguridad y de recuperación fiables?	1
2	Requiere comunicación de datos?	5
3	Requiere funciones de procesamiento distribuido?	0
4	Es critico el rendimiento?	2
5	Será ejecutado el sistema en un entorno operativo existente y fuertemente utilizado?	3
6	Requiere el sistema entrada de datos interactiva ?	1
7	Las transacciones de entrada se llevan a cabo sobre múltiples pantallas u operaciones?	0
8	Se utilizan archivos maestros de forma interactiva ?	2

Estimación de los puntos de función

9	Son complejas las entradas, las salidas, los archivos o las consultas ?	1
10	Es complejo el procesamiento interno ?	1
11	Se ha diseñado el código para ser reutilizable ?	4
12	Están incluidas en el diseño la conversión y la instalación ?	3
13	Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	3
14	Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente usada por el usuario?	4
Suma=		30

Valores de Ajuste

$$VAF = 0,65 + 0,01 * Suma = >$$

0,95

Estimación de primer orden de Jones

Si se tiene la suma total de todos los puntos de función, se puede realizar a partir de ellos un cálculo aproximado de la planificación, utilizando un método que Capers Jones ha denominado “Estimación de primer orden”.

Para utilizarlo, simplemente hay que tomar el total de los puntos de función y elevarlo a la potencia apropiada

Estimación de primer orden de Jones

Exponentes para calcular planificaciones a partir de puntos de función

Clase de Software	Mejor caso	Media	Peor caso
Sistemas	0,43	0,45	0,48
Gestión	0,41	0,43	0,46
“Pret-a-porter”	0,39	0,42	0,45

Estimación de primer orden de Jones

Planificaciones lo más cortas posibles

Tamaño del sistema LDC	Productos de Sistemas		Productos de Gestión		Productos "Pret-a-porter"	
	Planificación (Meses)	Esfuerzo (Pers.-Mes)	Planificación (Meses)	Esfuerzo (Pers.-Mes)	Planificación (Meses)	Esfuerzo (Pers.-Mes)
10.000	6,0	25,0	3,5	5,0	4,2	8,0
15.000	7,0	40,0	4,1	8,0	4,9	13,0
20.000	8,0	57,0	4,6	11,0	5,6	19,0
25.000	9,0	74,0	5,1	15,0	6,0	24,0
30.000	9,0	110,0	5,5	22,0	7,0	37,0
35.000	10,0	130,0	5,8	26,0	7,0	44,0
40.000	11,0	170,0	6,0	34,0	7,0	57,0
45.000	11,0	195,0	6,0	39,0	8,0	66,0
50.000	11,0	230,0	7,0	46,0	8,0	79,0
60.000	12,0	285,0	7,0	57,0	9,0	98,0
70.000	13,0	350,0	8,0	71,0	9,0	120,0
80.000	14,0	410,0	8,0	83,0	10,0	140,0
90.000	14,0	480,0	9,0	96,0	10,0	170,0
100.000	15,0	540,0	9,0	110,0	11,0	190,0

Estimación de primer orden de Jones

Planificaciones eficientes

Tamaño del sistema LDC	Productos de Sistemas		Productos de Gestión		Productos "Pret-a-porter"	
	Planificación (Meses)	Esfuerzo (Pers.-Mes)	Planificación (Meses)	Esfuerzo (Pers.-Mes)	Planificación (Meses)	Esfuerzo (Pers.-Mes)
10.000	8,0	24,0	4,9	5,0	5,9	8,0
15.000	10,0	38,0	5,8	8,0	7,0	12,0
20.000	11,0	54,0	7,0	11,0	8,0	18,0
25.000	12,0	70,0	7,0	14,0	9,0	23,0
30.000	13,0	97,0	8,0	20,0	9,0	32,0
35.000	14,0	120,0	8,0	24,0	10,0	39,0
40.000	15,0	140,0	9,0	30,0	10,0	49,0
45.000	16,0	170,0	9,0	34,0	11,0	57,0
50.000	16,0	190,0	10,0	40,0	11,0	67,0
60.000	18,0	240,0	10,0	49,0	12,0	83,0
70.000	19,0	290,0	11,0	61,0	13,0	100,0
80.000	20,0	345,0	12,0	71,0	14,0	120,0
90.000	21,0	400,0	12,0	82,0	15,0	140,0
100.000	22,0	450,0	13,0	93,0	15,0	160,0

Estimación de primer orden de Jones

Planificaciones nominales

Tamaño del sistema LDC	Productos de Sistemas		Productos de Gestión		Productos "Pret-a-porter"	
	Planificación (Meses)	Esfuerzo (Pers.-Mes)	Planificación (Meses)	Esfuerzo (Pers.-Mes)	Planificación (Meses)	Esfuerzo (Pers.-Mes)
10.000	10,0	48,0	6,0	9,0	7,0	15,0
15.000	12,0	76,0	7,0	15,0	8,0	24,0
20.000	14,0	110,0	8,0	21,0	9,0	34,0
25.000	15,0	140,0	9,0	27,0	10,0	44,0
30.000	16,0	185,0	9,0	37,0	11,0	59,0
35.000	17,0	220,0	10,0	44,0	12,0	71,0
40.000	18,0	270,0	10,0	54,0	13,0	88,0
45.000	19,0	310,0	11,0	61,0	13,0	100,0
50.000	20,0	360,0	11,0	71,0	14,0	115,0
60.000	21,0	440,0	12,0	88,0	15,0	145,0
70.000	23,0	540,0	13,0	105,0	16,0	175,0
80.000	24,0	630,0	14,0	125,0	17,0	210,0
90.000	25,0	730,0	15,0	140,0	17,0	240,0
100.000	26,0	820,0	15,0	160,0	18,0	270,0

Estimación del esfuerzo – COCOMO II

COCOMO II es un modelo de estimación de costo utilizable en la evaluación del esfuerzo estimado para un proyecto. Los principales objetivos del COCOMO II son:

- ❑ El modelo COCOMO.II completo incluye tres etapas.
 - ✓ La etapa 1 soporta la estimación de los esfuerzos de prototipado o desarrollo de aplicaciones.
 - ✓ La etapa 2 soporta la estimación a comienzos de la etapa de diseño de un proyecto, cuando se conoce poco acerca del costo del proyecto.
 - ✓ La etapa 3 soporta la estimación en la etapa post arquitectural del proyecto. La versión actual de COCOMO II implementa fórmulas de la etapa 3 para estimar el esfuerzo, tiempo, y costo requerido para desarrollar un producto software

Estimación del esfuerzo – COCOMO II

Entradas de COCOMO II

La entrada principal de COCOMO.II es el tamaño del programa, en KLOC, puntos de función o puntos de casos de uso. Se deben evaluar 17 atributos adicionales, estos atributos están incluidos en 4 categorías como sigue:

- ❑ **Atributos del Producto:** Estos atributos describen el ambiente en el cual opera el programa. Los atributos en esta categoría son: requerimientos de confiabilidad, tamaño de base de datos, concordancia entre la documentación y las necesidades del ciclo de vida, reutilización requerida y complejidad del programa.
- ❑ **Atributos de la Plataforma:** Estos atributos se refieren a las limitaciones que afectan el esfuerzo del desarrollo debido al hardware y al sistema operativo que se están utilizando en la ejecución del proyecto. Los atributos en esta categoría son restricciones de tiempo de ejecución, almacenamiento principal, y volatilidad de la plataforma.

Estimación del esfuerzo – COCOMO II

- ❑ **Atributos del Personal:** Estos atributos describen la habilidad del personal asignado al proyecto. Los atributos en esta categoría incluyen: capacidad del analista, experiencia en aplicaciones, capacidad del programador, experiencia en el lenguaje de programación, experiencia en la plataforma, continuidad del personal.
- ❑ **Atributos del Proyecto:** Estos atributos se refieren a las restricciones y condiciones bajo las cuales opera el desarrollo del proyecto. Los atributos en esta categoría son el uso de herramientas de desarrollo de software y desarrollo multi-sitio.

Estos factores (o multiplicadores de esfuerzo) se multiplican y de esta forma están incorporados en las fórmulas de estimación del tiempo y esfuerzo. El valor numérico del i -ésimo factor de ajuste es llamado EM_i y su producto es llamado factor de ajuste o EAF. El esfuerzo total PM_{total} es el producto del esfuerzo nominal y el EAF.

Estimación del esfuerzo – COCOMO II

Procesamiento COCOMO II

Utilizando COCOMO II, una evaluación nominal de los meses-hombre basada sólo en el tamaño se realiza al programa considerado. A continuación, se multiplica la evaluación de todos los atributos para obtener el esfuerzo en meses-hombre requerido por el proyecto. Los desafíos principales al utilizar COCOMO II son determinar el tamaño del proyecto y asignar los valores apropiados a los atributos.

Estimación del esfuerzo – COCOMO II

Salidas de COCOMO II

La salida del modelo COCOMO II es simplemente el nivel de esfuerzo en meses-hombre para el proyecto siendo estimado y el tiempo en meses. El esfuerzo puede fácilmente ser convertido a valor monetario si el costo por mes-hombre es conocido. La Distribución de Fases es una de las salidas. Su función es la de mostrar un desglose del esfuerzo del software y el tiempo requerido a las fases del ciclo de desarrollo. Estas fases son análisis de requerimientos, diseño, implementación e integración y prueba. Las salidas del modelo son muy básicas y no muy flexibles, por lo tanto las métricas de rendimiento deberán ser creadas fuera de este modelo.

Estimación del esfuerzo – COCOMO II

Calibración COCOMO II

La calibración es esencial para el uso correcto de modelos de costo de software. El usuario de COCOMO II puede calibrar los EAF y las ecuaciones de esfuerzo/tiempo del proyecto actual.

Estimación del esfuerzo – COCOMO II

Aplicabilidad de la métrica

El método de estimación COCOMO II está basado dos modelos: uno aplicable al comienzo de los proyectos (Diseño preliminar, en inglés Early Design) y otro aplicable luego del establecimiento de la arquitectura del sistema (Post arquitectura, en inglés Post Architecture). El modelo de Diseño preliminar (Early Design) contempla la exploración de las arquitecturas alternativas del sistema y los conceptos de operación. En esta etapa no se sabe lo suficiente del proyecto como para hacer una estimación fina. Ante ésta situación, el modelo propone la utilización de Puntos de Función como medida de tamaño y un conjunto de 7 factores (cost drivers) que afectan al esfuerzo del proyecto. Estos 7 factores son agrupaciones de los factores que se utilizan en la otra variante del modelo (Post Arquitectura).

Estimación del esfuerzo – COCOMO II

El modelo Post arquitectura (Post Architecture) contempla el desarrollo y el mantenimiento de un producto software. Esta estrategia es más precisa si se ha desarrollado una arquitectura del sistema, la cual haya sido validada y establecida como base para la evolución del producto. Ante ésta situación, el modelo propone la utilización de Líneas de código fuente y/o Puntos de Función como medidores del tamaño, modificadores para indicar el grado de reutilización y descarte del software, un conjunto de 17 estimadores de costo, y un conjunto de 5 factores que afectan de manera exponencial en el esfuerzo del proyecto.

Estimación del esfuerzo – COCOMO II

Detalles de la métrica

Tanto en el modelo de diseño preliminar como en el post arquitectural, la estimación del esfuerzo se realiza tomando como base la siguiente ecuación:

$$PM_{\text{nominal}} = A \times (\text{Size})^B$$

Donde:

- ❑ **PM nominal:** es el esfuerzo nominal requerido en meses- hombre
- ❑ **Size:** es el tamaño estimado del software, en miles de líneas de código (KLDC) o en Puntos de Función sin ajustar (convertibles a KLDC mediante un factor de conversión que depende del lenguaje y la tecnología).

Estimación del esfuerzo – COCOMO II

Valoración del Factor escalar B

El factor escalar B se calcula a partir de la sumatoria de los aportes de distintas Variables escalares, las cuales son variables que indican las características que el proyecto presenta en lo que a su complejidad y entorno de desarrollo se refiere. Las Variables escalares de COCOMO II son las siguientes:

- ☐ PREC, variable de precedencia u orden secuencial del desarrollo
- ☐ FLEX, variable de flexibilidad del desarrollo
- ☐ RSEL, indica la fortaleza de la arquitectura y métodos de estimación y reducción de riesgos
- ☐ TEAM, esta variable refleja la cohesión y madurez del equipo de trabajo
- ☐ PMAT, relaciona el proceso de madurez del software

[illegible]

Estimación del esfuerzo – COCOMO II

Luego de la ponderación de éstas variables, el Factor escalar se calcula mediante la siguiente ecuación:

$$B = 0.91 + 0.01 \times \Sigma(W_i)$$

Estimación del esfuerzo – COCOMO II

Ajuste del esfuerzo nominal

Los Multiplicadores de esfuerzo se cuantifican con una escala que va desde Extra Bajo a Extra Alto, y cada multiplicador tiene un valor asociado a cada nivel de la escala. Cada uno de los modelos de estimación (Diseño preliminar y Post arquitectura) tiene un conjunto de Multiplicadores de esfuerzo, los cuales son acordes con la información que se maneja en cada uno de estos modelos.

En ambos modelos, el esfuerzo ajustado se calcula mediante la siguiente ecuación:

$$PM_{\text{ajustado}} = PM_{\text{nominal}} \times \prod(EM_i)$$

Estimación del esfuerzo – COCOMO II

Multiplicadores de esfuerzo en el modelo Post arquitectura

- ❑ Para este modelo, los multiplicadores son 17, agrupados en las siguientes categorías: producto, plataforma, personal y proyecto. A continuación se muestran los multiplicadores, con una breve descripción de su significado.
- ❑ Multiplicadores que afectan al producto:
RELY: Confiabilidad requerida del software. Mide el impacto que tiene una falla en el software.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto
RELY	Inconvenientes imperceptibles	Bajo, y con pérdidas fácilmente recuperables	Moderado, con pérdidas de fácil recuperación.	Altas pérdidas Financieras	Riesgo para la Vida humana.

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

DATA: Tamaño de la base de datos. Se mide como el tamaño de la base en bytes sobre el tamaño del programa en LOC. Se utiliza para dimensionar el esfuerzo requerido para el control y la generación de datos de prueba.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto
DATA		$D/P < 10$	$10 \leq D/P < 100$	$100 \leq D/P < 1000$	$D/P \geq 1000$

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

CPLX: Complejidad del producto. La complejidad se divide en cinco áreas: Operaciones de Control, operaciones de Cálculo, Dependencia de Dispositivos, Manejo de Datos e Interfaces de Usuario.

	Operaciones de Control	Operaciones de Cálculo	Dependencia de Dispositivos	Manejo de Datos	Interfaces de Usuarios
Muy Bajo	Programación lineal, con muy pocas estructuras no anidadas: DO, IF, CASE. Composición de módulos simples a través de procedimientos y funciones	Evaluaciones de expresiones simples.	Escrituras y grabaciones con formatos simples.	Manejo de vectores simples en memoria. Manejo de consultas y accesos a la base de datos en forma sencilla.	Formularios de ingreso sencillos. Generación de reportes simples.
Bajo	Estructuras bien anidadas	Evaluaciones moderadas de cálculos. Ej. Raíz cuadrada, exponentes	Sin particularidades o dependencias del procesador de E/S. Se manejan a través de GET y PUT de conjuntos de datos	Manejo de datos sin archivos intermedios, sin edición. COTS-DB, consultas y actualizaciones moderadas	Uso de GUI (grafic user interface, interfaces graficas de usuario) simples
Nominal	La mayoría de las estructuras son anidadas. Con controles de intercambio simple de datos entre módulos a través de parámetros, tablas de decisiones, soporte para procesamiento distribuido de baja complejidad	Uso de rutinas básicas y estándar. Manejo básico de vectores y matrices	Operaciones de E/S que permiten seleccionar el dispositivo, haciendo un control del estado de los mismos y procesando errores	Ingreso con formatos múltiples de datos, pero conservando un único formato de salida. Cambios estructurales simples con ediciones. Uso de consultas y COTS-DB complejos	Uso simple de un conjunto de parámetros de definición de interfaz del usuario.
Alto	Total uso de estructuras anidadas. Manejo de pilas y colas. Desarrollos para un único procesador	Análisis numérico sencillo: interpolación, ecuaciones diferenciales. Uso de redondeo y trunc.	Operaciones de E/S a niveles físicos usando direccionamiento para la lectura y búsqueda. Overlap optimizado para estas operaciones	Activación de disparadores a través de datos de la DB. Reestructuraciones complejas de datos.	Uso de parámetros de definición de interfaces. Uso simple de características multimediales (entrada a través de la voz)
Muy Alto	Código recursivo. Manejo de interrupciones, y sincronización de tareas complejas. Procesamiento distribuido heterogéneo. Un único procesador controlado en tiempo real	Cálculos numéricos difíciles pero estructurados: ecuaciones matriciales, diferenciales	Rutinas para el diagnóstico de interrupciones. Manejo de la línea de comunicación entre dispositivos. Uso intensivo del manejo de performance	Uso complejo de disparadores. Optimización de consultas. Coordinación de bases de datos distribuidas	Uso moderado de 2 y 3 dimensiones. Habilidades gráficas complejas y multimediales.
Extra alto	Operaciones de control múltiples con cambios dinámicos de prioridades. Control a nivel microcódigo. Control en tiempo real del hardware distribuido.	Análisis numérico complejo y no estructurado. Cálculos complejos en paralelo	Codificación de dispositivos asincrónica, con control crítico de performance y procesos.	Uso de lenguajes de manejo de datos, y técnicas de objetos así como relacionales	Uso complejo de multimedia y realidad virtual

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

RUSE: Reusabilidad del código. Mide el costo adicional requerido para diseñar componentes más genéricos, mejor documentados y más confiables, de manera a reutilizarlos en otros proyectos.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
RUSE		Nada.	Por proyecto.	Por programa	Por línea de producto.	Por múltiples líneas de producto.

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

DOCU: Documentación. Evalúa los requerimientos de documentación a lo largo del ciclo de vida del proyecto.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto
DOCU	Muchas etapas del ciclo de vida están sin documentación.	Algunas.	De acuerdo a las necesidades exactas de las etapas del ciclo de vida.	Excesiva.	Muy excesiva.

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

STOR: Restricciones de almacenamiento principal. Similar al multiplicador anterior, pero relacionadas con el espacio principal de almacenamiento.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
STOR			$\leq 50\%$ de uso de los recursos disponibles	70 %	85 %	95 %

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

PVOL: Volatilidad de la plataforma. Expresa la velocidad de cambio del hardware y el software usados como plataforma.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PVOL		Grandes Cambios c/12 meses Cambios menores c/ 1 mes	Grandes Cambios c/6 meses Cambios menores c/ 2 semanas	Grandes Cambios c/2 meses Cambios menores c/ 1 semana	Grandes Cambios c/ 2 semanas Cambios menores c/ 2 días	

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

ACAP: Capacidad de los analistas. Se considera la capacidad de análisis y diseño, eficiencia, habilidad para comunicarse y trabajar en equipo. No se considera el nivel de experiencia.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
ACAP	15 %	35 %	55 %	75 %	90 %	

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

PCAP: Capacidad de los programadores. Se considera la capacidad de trabajo en equipo, eficiencia y habilidad para comunicarse. No se considera el nivel de experiencia.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PCAP	15 %	35 %	55 %	75 %	90 %	

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

AEXP: Experiencia en aplicaciones. Contempla el nivel de experiencia del grupo de desarrollo (principalmente analistas) en aplicaciones equivalentes.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
AEXP	2 meses	6 meses	1 año	3 años	6 años	

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

PEXP: Experiencia en la plataforma. Refleja la experiencia del grupo de desarrollo (principalmente programadores) en el uso de herramientas de software y hardware utilizado como plataforma.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PEXP	2 meses.	6 meses	1 año.	3 años	6 años	

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

LTEX: Experiencia en el lenguaje y herramientas de desarrollo. Refleja la experiencia del grupo de desarrollo en el lenguaje de programación y las herramientas de desarrollo utilizadas.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
LTEX	2 meses	6 meses	1 año	3 años	6 años.	

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

PCON: Continuidad del personal. Expresa el porcentaje de rotación anual del personal afectado al proyecto.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PCON	48% al año	24% al año	12% al año	6% al año	3% al año	

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

TOOL: Uso de herramientas de software. Contempla el uso de herramientas, desde la edición hasta el manejo de todo el ciclo de vida.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto
TOOL	Edición y codificación con debug	CASE simple y de poca integración	Herramientas básicas para todo el ciclo de vida con moderada integración	Potentes herramientas a ser usadas en todo el ciclo de vida con integración moderada	Herramientas potentes y preactivas, muy bien integradas con el proceso, los métodos y la reusabilidad

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

SITE: Desarrollo en múltiples ubicaciones. Involucra la ubicación física y el soporte de comunicaciones.

Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Algo de teléfono y mail	Fax y teléfonos individuales	Red de correo electrónico interno	Comunicaciones electrónicas que cubren todas las ubicaciones	Comunicaciones electrónicas que cubren todas las ubicaciones con la posibilidad de videoconferencias ocasionales.	Multimedia

Estimación del esfuerzo – COCOMO II

Multiplicadores que afectan al producto:

SCED: Requerimientos de calendario de desarrollo. Refleja las restricciones impuestas al grupo de desarrollo sobre la agenda nominal estimada del proyecto.

	Muy bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
SCED	75% del nominal.	85 %	100 %	130 %	160 %	

Estimación del esfuerzo – COCOMO II

Multiplicadores de esfuerzo en el modelo de Diseño preliminar:

Para este modelo, los multiplicadores son 7, y se obtienen como combinaciones de los multiplicadores del modelo Post arquitectura. Estos multiplicadores son:

PERS: Capacidad del personal. Está dado por la suma o la combinación porcentual de los multiplicadores ACAP, PCAP y PCON.

	Extra Bajo	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Suma de ACAP, PCAP, PCON	3,4	5,6	7,8	9	10,11	12,13	14,15
Combinación de ACAP y PCAP	20%	39%	45%	55%	65%	75%	85%
Rotación anual del personal	45%	30%	20%	12%	9%	5%	4%

Estimación del esfuerzo – COCOMO II

Multiplicadores de esfuerzo en el modelo de Diseño preliminar:

RCPX: Complejidad del producto. Está dado por la combinación de los multiplicadores RELY, DATA, CPLX y DOCU.

	Extra Bajo	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Suma de RELY, DATA, CPLX y DOCU	5,6	7,8	9-11	12	13-15	16-18	19-21
Énfasis en la documentación	Muy poca	Poca	Algo	Básica	Fuerte	Muy fuerte	Extrema
Complejidad del producto	Muy simple	Simple	Algo	Moderada	Compleja	Muy compleja	Extremadamente compleja
Tamaño de la base de datos	Pequeña	Pequeña	Pequeña	Moderada	Grande	Muy Grande	Muy Grande

Estimación del esfuerzo – COCOMO II

Multiplicadores de esfuerzo en el modelo de Diseño preliminar:

- ❑ RUSE: Reusabilidad. Está dado por el mismo multiplicador RUSE del modelo Post arquitectura.
- ❑ PDIF: Dificultad de la plataforma. Está dado por la combinación de los multiplicadores TIME, STOR y PVOL.
- ❑ PREX: Experiencia del personal. Está dado por la combinación de los multiplicadores AEXP, PEXP y LTEX.
- ❑ SCED: Calendario. Está dado por el mismo multiplicador SCED del modelo Post arquitectura.
- ❑ FCIL: Facilidades. Está dado por la combinación de los multiplicadores TOOL y SITE.

Estimación del esfuerzo – COCOMO II

Ecuación de tiempo (schedule equation)

Predice el número de meses requeridos para completar el proyecto. La duración del proyecto está basada en el esfuerzo calculado por la ecuación del esfuerzo:

$$\text{Duración} = 3.67 * (PM_{\text{ajustado}})^{[0.28 + 0.2 * (B-0.91)]}$$

Estimación del esfuerzo – COCOMO II

Ejemplo de Aplicación del modelo

Aplicación sobre los Puntos de Función sin ajustar

- ❑ Éste método es el preferido en la actualidad para la estimación del esfuerzo cuando no se tiene información histórica a la cual recurrir.
- ❑ Consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar o la cantidad de líneas de código (LDC, Líneas de Código) estimados para un proyecto.
- ❑ A manera de ejemplo, consideremos un proyecto que tiene el siguiente valor para los puntos de función sin ajustar: UFP = 26
- ❑ Para aplicar la ecuación de cálculo del esfuerzo nominal, necesitamos convertir los puntos de función sin ajustar a KLDC y por otro calcular el Factor escalar B de acuerdo a las características del proyecto. Luego:

$$PM_{\text{nominal}} = A \times (\text{Size})^B$$

Estimación del esfuerzo – COCOMO II

Donde:

- ❑ **A:** tomamos el valor por defecto del modelo, ajustado en 2.94
- ❑ **Size:** se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Supongamos que utilizamos Java (factor de conversión = 53 LDC/UFP).

Entonces tendremos:

$$\text{Size} = 53 \times 26 = 1378 \text{ LDC}$$

- ❑ **B:** se calcula ponderando las variables escalares mediante la ecuación:

$$B = 0.91 + 0.01 \times \Sigma(W_i)$$

Estimación del esfuerzo – COCOMO II

Donde:

- ❑ A: tomamos el valor por defecto del modelo, ajustado en 2.94

Size: se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema. Supongamos que utilizamos Java (factor de conversión = 53 LDC/UFP).

Entonces tendremos:

$$\text{Size} = 53 \times 26 = 1378 \text{ LDC}$$

- ❑ B: se calcula ponderando las variables escalares mediante la ecuación:

$$B = 0.91 + 0.01 \times \Sigma(W_i)$$

Estimación del esfuerzo – COCOMO II

Donde se considerarán los valores de las W_i mostradas en la siguiente tabla:

Variable	Descripción	Ponderación	Valor
PREC	El sistema es muy familiar	Muy Alto	1.24
FLEX	Algo de relajación en cuanto a la flexibilidad del desarrollo	Nominal	3.04
RESL	La arquitectura es sólida y los riesgos generalmente se mitigan	Alto	2.83
TEAM	La interacción del equipo es altamente cooperativa	Muy Alto	1.10
PMAT	La madurez del proceso software es baja	Bajo	6.24
Total			1.05

Estimación del esfuerzo – COCOMO II

Finalmente, el esfuerzo nominal resulta:

$$PM_{\text{nominal}} = A \times (\text{Size})^B = 2.94 \times (1.378)^{1.05} = 4.11 \text{ Meses-hombre}$$

Para completar la estimación, hay que ajustar el esfuerzo nominal de acuerdo a las características del proyecto. El ajuste se efectúa aplicando la ecuación:

$$PM_{\text{ajustado}} = PM_{\text{nominal}} \times \prod (ME_i)$$

Donde los ME_i (multiplicadores de esfuerzo) varían en función del modelo de estimación seleccionado (Diseño Preliminar o Post arquitectura).

Estimación del esfuerzo – COCOMO II

En nuestro caso vamos a aplicar el modelo de Diseño preliminar. Entonces, cuantificamos los multiplicadores de esfuerzo para éste modelo:

Multiplicador	Descripción	Ponderación	Valor
PERS	Se tienen analistas y programadores con alta eficiencia y capacidad de trabajo en equipo. Dedicación full-time.	Alto	0.83
RCPX	Las exigencias de confiabilidad, documentación y volumen de datos son moderadas, y la complejidad del producto es baja.	Nominal	1
RUSE	No se pretende reutilizar nada	Bajo	0.95
PDIF	No existen restricciones en cuanto al tiempo de CPU o al consumo de memoria, la plataforma es muy estable.	Bajo	0.87
PREX	Tanto los analistas como los programadores tienen aproximadamente 6 meses de experiencia en la aplicación, la plataforma, el lenguaje y las herramientas utilizadas.	Muy Bajo	1.33
SCED	Se requiere terminar el proyecto en el tiempo estimado.	Nominal	1
FCIL	Se tienen herramientas CASE simples e infraestructura de comunicaciones básica.	Bajo	1.10
Total			1.004

Estimación del esfuerzo – COCOMO II

Con estos valores, el ajuste del esfuerzo resulta:

$$PM_{\text{ajustado}} = 4.11 \times 1.004 = 4.13 \text{ Meses-hombre}$$

Expresando el mismo valor en Horas-hombre, y teniendo en cuenta que un mes es aproximadamente 160 horas, el esfuerzo resulta:

$$PM_{\text{ajustado}} = 4.13 \times 160 = 660.8 \text{ Horas-hombre}$$

$$\text{Duración} = 3.67 * (4.13)^{[0.28 + 0.2 * (B-0.91)]} = 5.6 \text{ meses}$$

Estimación del esfuerzo – COCOMO II

Extensiones:

- ☐ COCOTS (Constructive COTS)
- ☐ CORADMO (Constructive Rapid Application Development Model)
- ☐ COQUALMO (Constructive Quality Model)
- ☐ COSEMO (Constructive Staged Schedule an Effort Model)
- ☐ COPROMO (Constructive Productivity improvement Model)

Estimación del esfuerzo – COCOMO II

Herramientas:

- ☐ USC COCOMO II
- ☐ COSTAR

Sitio Web :

http://sunset.usc.edu/research/COCOMOII/expert_cocomo/expert_cocomo2000.html



MUCHAS GRACIAS