



**Universidad
Israel**

Ingeniería de Software II

Tema: Estimación de Proyectos de Software



Docente: *Mg. Mario Ruben Pérez Cargua*



Universidad
Israel

Objetivo

Aplicar los diferentes métodos que existen para estimar costos a los proyectos de software

Contenido

- Estimar Proyectos
- Métodos mas Utilizados



**Universidad
Israel**

- Porque Estimar Proyectos

Estimación Proyectos de Software

Antecedentes

Las mejoras en el proceso de desarrollo de software y sistemas de calidad no pueden ser evaluadas sin un esfuerzo efectivo de medición.

Necesidades de medición

- Mejoras en la calidad y productividad.
- Planificación y estimación de proyectos con alguna precisión.
- Disposición del personal adecuado, bien utilizado y motivado.
- Existencia de una adecuada estructura organizacional.
- Uso de técnicas y herramientas efectivas para el proceso.
- Obtención de un espacio físico y ambiente de trabajo óptimo.

Estimación Proyectos de Software

Antecedentes

Los aspectos esenciales de la medición son:

- **Datos duros**, son cuantificables con poca o sin subjetividad (esfuerzo, volumen documentación, errores detectados, etc.)
- **Datos blandos**, presentan un grado de subjetividad (habilidad y experiencia, presiones de tiempo, satisfacción del cliente, cooperación del cliente, etc.)
- **Datos normalizados**, son usados con propósito comparativo (LOC, PF, CC, PO)

Estimación Proyectos de Software

Escalas de dimensionamiento

- **Escala Nominal**, la cual ordena ítems por categoría. Un ejemplo de esta escala de medición es cuando catalogamos un lenguaje de programación: C++, Java, entre otros.
- **Escala Ordinal**, la cual ordena ítems. Un ejemplo es cuando se asigna una severidad a la falla encontrada como menor, mayor, catastrófica.
- **Escala de Intervalos**, la cual define una distancia desde un punto a otro. Este tipo de escala entrega cálculos no disponibles en la escala ordinal, como el cálculo del significado
- **Escala de Proporción**, esta escala es la que entrega mayor información y flexibilidad, debido a que incorpora el cero absoluto. Mediciones como LOC y número de defectos son mediciones de proporción.

Estimación Proyectos de Software

Que se puede medir en el Software

En general se desea medir los siguientes aspectos en Ingeniería de Software:

- **Procesos o tareas a ejecutar** (modelado, diseño, prueba).
- **Productos entregados durante el proceso** (documentación de diseño, código fuente, registro de pruebas).
- **Recursos que permiten realizar el proceso** (personal, computadoras, dinero).

Estimación Proyectos de Software

Que se puede medir en el Software

En general se desea medir los siguientes aspectos en Ingeniería de Software:

- **Procesos o tareas a ejecutar** (modelado, diseño, prueba).
- **Productos entregados durante el proceso** (documentación de diseño, código fuente, registro de pruebas).
- **Recursos que permiten realizar el proceso** (personal, computadoras, dinero).

Atributos internos y externos

Un atributo interno es medido directamente desde la entidad. (código fuente : el tamaño medido por las líneas de código).

Un atributo externo es una medida de la entidad con relación a una necesidad externa definida por el ambiente en el cual es desarrollada o utilizada. Por ejemplo, la mantenibilidad del código fuente

Estimación Proyectos de Software

Que se puede medir en el Software

PRODUCTO		
Entidades	Atributos Internos	Atributos Externos
Especificaciones	Tamaño, re-uso, modularidad, redundancia, funcionalidad, correctitud	Comprensibilidad, mantenibilidad
Diseño	Tamaño, re-uso, modularidad, acoplamiento, cohesividad, funcionalidad	Calidad, complejidad, mantenibilidad
Codificación	Tamaño, re uso, modularidad, acoplamiento, funcionalidad, complejidad algorítmico, estructuración	Confiabilidad, Usabilidad, mantenibilidad
Datos de prueba	Tamaño, nivel de cobertura	Calidad

Estimación Proyectos de Software

Que se puede medir en el Software

PROCESOS		
Entidades	Atributos Internos	Atributos Externos
Especificación de requerimientos	Tiempo, esfuerzo, número de cambios en los requerimientos	Calidad, costo, estabilidad
Diseño detallado	Tiempo, esfuerzo, número de especificaciones erróneas detectadas	Costo, costo - eficacia
Pruebas	Tiempo, esfuerzo, número y errores encontrados	Costo, costo - eficacia, estabilidad
Vista de Despliegue	Diagrama de Despliegue	Nodo, componente, dependencia, localización.

Estimación Proyectos de Software

Que se puede medir en el Software

RECURSOS		
Entidades	Atributos Internos	Atributos Externos
Personal	Años de experiencia, tasa de trabajo	Productividad, experiencia, inteligencia.
Equipos	Tamaño, nivel de comunicación, estructuración	Productividad, calidad.
Software	Tamaño, costo	Usabilidad, Confiabilidad
Hardware	Precio, velocidad, tamaño de la memoria	Confiabilidad
Oficinas	Tamaño, temperatura, luz	Confort, calidad

Estimación Proyectos de Software

Definición Atributos (Pg. 55)

Atributo	Definición
Acoplamiento	Grado de fuerza de la inter-conexión entre los componentes del sistema.
Calidad	Grado de excelencia.
Cohesividad	Grado en que las entidades dependen o no de las otras entidades.
Correctitud	Grado en el que una entidad satisface las especificaciones y cumple los objetivos del usuario.
Complejidad	Cantidad de diversos elementos que lo componen.
Complejidad algorítmica	Grado de relativa dificultad computacional de las funciones.
Costo	Cuanto cuesta conseguir un objetivo determinado.(Monetariamente)
Costo-eficiencia	Grado en que el costo esta relacionado con la eficiencia obtenida.
Confiabilidad	Grado con el que una entidad realiza su función con una precisión requerida.
Confort	Grado de comodidad.

Estimación Proyectos de Software

Algunos Indicadores de Medición

$$\text{Productividad} = \text{KLOC} / \text{PM}$$

KLOC son las Kilo Líneas de Código fuente entregada y PM son las Personas Mes destinados al proyecto.

$$\text{Calidad} = \text{defectos} / \text{KLOC}$$

$$\text{Costo} = \$ / \text{LOC}$$

$$\text{Documentación} = \text{páginas de documentación} / \text{KLOC}$$

Las mediciones asociadas a las LOC están en permanente discusión, ya que son dependientes del lenguaje de programación

Estimación Proyectos de Software

Atributos de las técnicas de estimación

La estimación de costos, tiempos y recursos para el esfuerzo de desarrollo de Software **requiere experiencia**, acceso a una **buena información histórica** y coraje para **confiar en medidas cuantitativas** cuando todo lo que existe son datos cualitativos.

- **La complejidad del proyecto**, que tiene un gran efecto sobre la incertidumbre que es inherente a la planificación.
- **El tamaño del proyecto**, que es otro factor importante y puede afectar a la precisión y eficacia de las estimaciones
- **El grado de estructuración del proyecto**, que también tiene efecto sobre el riesgo de la estimación

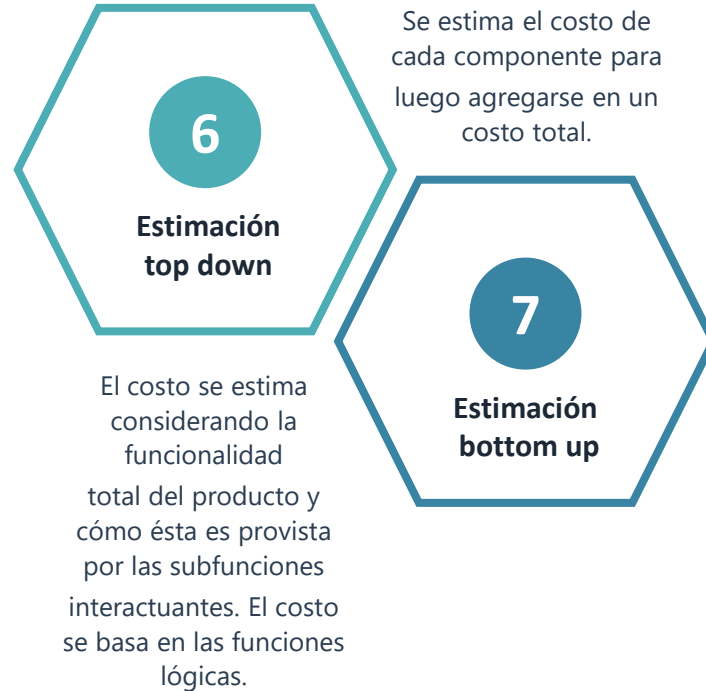
Estimación de Costos

Técnicas para la Estimación de Costos



Estimación de Costos

Técnicas para la Estimación de Costos



Los factores que afectan a la estimación de un proyecto son básicamente dos: la complejidad del proyecto y el tamaño del mismo.

Estimación Proyectos de Software

Estimación de Costos

Los componentes principales de costos son:

- Hardware.
- Entrenamiento.
- Esfuerzo.



**Universidad
Israel**

- Modelo Punto Función

Métricas del Software

Método Punto Función

El Análisis de Puntos de Función (FPA, por sus siglas en inglés) es una **técnica de medición** de las funcionalidades ofrecidas por un **software** desde el punto de vista de sus **usuarios**.

Punto de función (FP, por sus siglas en inglés) , que es su unidad de medida, tiene por objetivo tornar la medición independiente de la tecnología utilizada para su construcción. Es decir, el FPA busca medir lo que el software hace y no como es construido (No considera ningún aspecto de implementación).

Método estándar ISO/IEC 20926 de medición de software que cuantifica los requisitos funcionales del usuario

No por tener mas pantallas el sistema tiene que costar mas

Métricas del Software

Método Punto Función

- Estimar el esfuerzo en un proyecto informático horas hombre (HH)
- Estimar la duración de un proyecto de software (en meses)
- Estimar el costo del proyecto informático

Análisis divide la especificación funcional

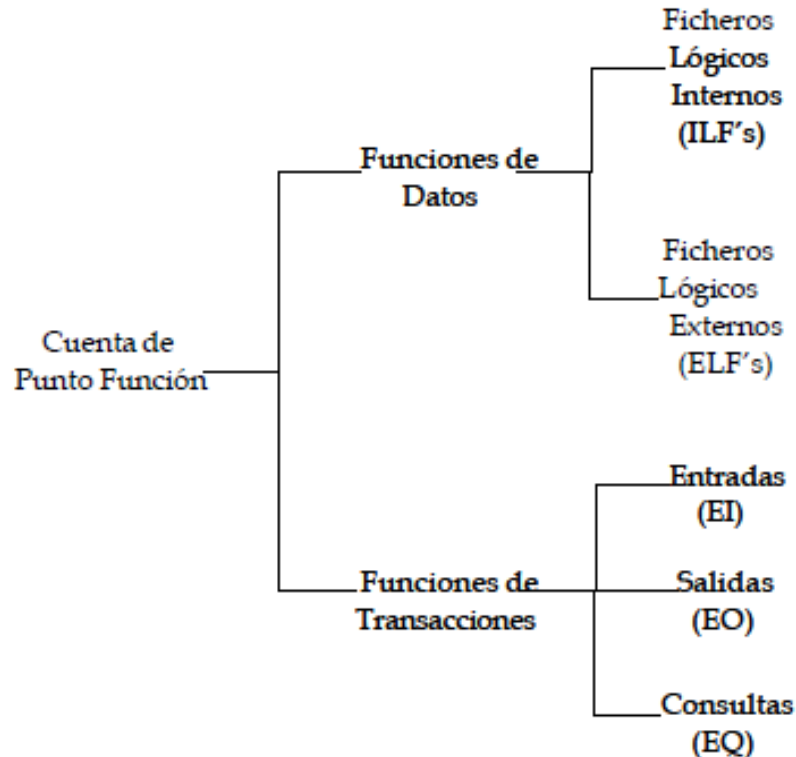
- Interacción
- Almacenamiento

Componentes funcionales básicos

- Interacción Función de transacción (interacciones con el usuario)
- Almacenamiento Función de datos

Métricas del Software

Método Punto Función



Almacenamiento Función de datos

Archivo lógico interno (ILF : Internal Logical File)

- Archivo del punto de vista lógico, no como en un sistema operativo
- Puede ser tabla en la base de datos

Archivo lógico externo (ELF : External Interface File)

- Datos referenciados a otros sistemas
- Datos mantenidos por otros sistemas, pero usados por el sistema actual

Interacción Función de transacción

Entrada externa (EI : External input)

- Pantallas donde el usuario ingresa datos

Salida externa (EO : External output)

- Informes, gráficos, Lista de datos

Consulta externa (EQ : External query)

- Recuperar y mostrar datos al usuario - buscar

Métricas del Software

Método Punto Función

- Buscar (EQ)
- Actualizar (EI)
- Insertar (EI)
- Listar (EO)
- Eliminar (EI)
- Informes o reportes (EO)
- Tablas de BD (ILF)

Tipo / Complejidad	Baja	Media	Alta
(EI) Entrada externa	3 PF	4 PF	6 PF
(EO) Salida externa	4 PF	5 PF	7 PF
(EQ) Consulta externa	3 PF	4 PF	6 PF
(ILF) Archivo lógico interno	7 PF	10 PF	15 PF
(EIF) Archivo de interfaz externo	5 PF	7 PF	10 PF

Valores estándar (IFPUG) International Function Point Users Group

Métricas del Software

Método Punto Función

Requerimientos
Funcionales

- Para el siguiente ejemplo, se considerará que todas las funciones identificadas serán de **complejidad media**.
- El sistema requerido es:
 - Registro de Equipos de futbol
 - Registros de partidos
 - Buscar partido por fecha
 - Actualización de datos del equipo
 - Eliminar equipos
 - Listado de equipos
 - 1 reporte de los equipos registrados por rango de fechas
 - 1 reporte de partidos

Métricas del Software

Método Punto Función

- Registro de Equipos de futbol (EI 4 PF)
- Registros de partidos (EI 4 PF)
- Buscar partido por fecha (EQ 4 PF)
- Actualización de datos del equipo (EI 4 PF)
- Eliminar equipos (EI 4 PF)
- Listado de equipos (EO 5 PF)
- 1 reporte de los equipos registrados por rango de fechas (EO 5 PF)
- 1 reporte de partidos (EO 5 PF)
- 4 Tablas en BD (ILF 40 PF)

Puntos de función sin ajustar (PFSA): 75

Métricas del Software

Método Punto Función

Tipo / Complejidad	Baja	Media	Alta	TOTAL
(EI) Entrada externa	3 PF	4 x 4 PF	6 PF	16
(EO) Salida externa	4 PF	3 x 5 PF	7 PF	15
(EQ) Consulta externa	3 PF	1 x 4 PF	6 PF	4
(ILF) Archivo lógico interno	7 PF	4 x 10 PF	15 PF	40
(EIF) Archivo de interfaz externo	5 PF	0 x 7 PF	10 PF	0
			PFSA	75

Métricas del Software

Determinación de los niveles de influencia

Factor de Ajuste	Puntaje
Comunicación de Datos	4
Procesamiento Distribuido	4
Objetivos de Rendimiento	1
Configuración del equipamiento	1
Tasa de transacciones	3
Entrada de Datos en Línea	5
Interfase con el usuario	2
Actualizaciones en Línea	3
Procesamiento Complejo	1
Reusabilidad del Código	1
Facilidad de Implementación	
Facilidad de Operación	1
Instalaciones Múltiples	2
Facilidad de Cambios	4
Factor de Ajuste	32

El cálculo del factor de ajuste está basado en 14 características generales de los sistemas que miden la funcionalidad general de la aplicación. A cada característica se le atribuye un peso que varía de 0 a 5 e indica el grado o nivel de influencia que cada característica tiene en la aplicación que está siendo estudiada.

<https://www.oocities.org/gimenezpy/ajuste.htm>

Métricas del Software

Estimación del Esfuerzo

- $PFA = PFSA * [0.65 + (0.01 * \text{factor de ajuste})]$

- Donde:

- **PFSA**: Puntos de función sin ajustar
- **PFA** : Puntos de función ajustado

$$PFA = 75 * [0.65 + (0.01 * 32)]$$

$$PFA = 75 * [0.65 + 0.32]$$

$$PFA = 75 * 0.97$$

$$PFA = 73.8 \rightarrow 74$$

Estimación de esfuerzo (H/H)

Lenguaje	Horas PF promedio	Lineas de código por PF
Ensamblador	25	300
COBOL	15	100
Lenguajes 4ta Generación	8	20

$$H/H = PFA * \text{Horas PF promedio}$$

$$H/H = 74 * 8$$

$$H/H = 592 \text{ Horas hombre}$$

Ejemplo:

5 horas diarias de trabajo

1 mes = 20 días

$$592 / 5 = 118,4 \text{ días de trabajo}$$

118,4 / 20 = 5,92 meses para desarrollar el software de lunes a viernes 5 horas diarias con 1 trabajador (ESTIMACIÓN de duración del proyecto)

Métricas del Software

Estimación del Tiempo y Presupuesto

H/H = 592 Horas hombre

Desarrolladores = 3

Horas = $592 / 3 = 197,3$ horas (Duración del proyecto en horas)

$197,3 / 5 = 39,46$ días de trabajo

$39,46 / 20 = 1,97$ meses para desarrollar el software de lunes a viernes 5 horas diarias con 3 desarrolladores (ESTIMACIÓN de duración del proyecto)

Costo = (Desarrolladores * Duración del proyecto en meses * sueldos)+ otros

Sueldos=

Duración del proyecto en meses= 1.97

Otros=

Costo Estimado:



**Universidad
Israel**

- Modelo COCOMO

Estimación Proyectos de Software

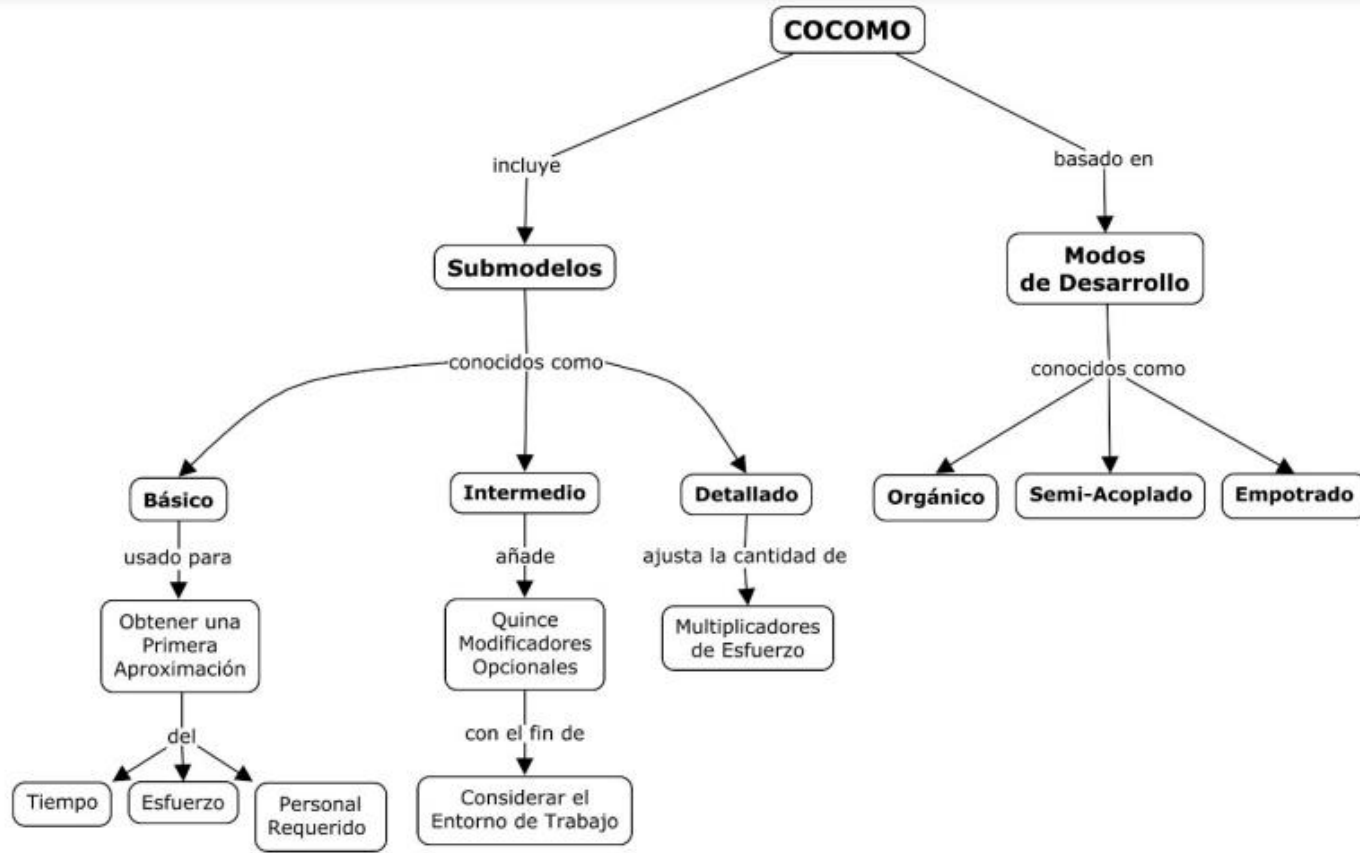
Método COCOMO

Modelo COCOMO II, modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste).

El modelo COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados en la industria.

Estimación Proyectos de Software

Estimación del Esfuerzo



Estimación Proyectos de Software

Método COCOMO II

Los submodelos son tres: básico, intermedio y detallado. Por su parte, los modos de desarrollo son también tres: orgánico, semi-acoplado y empotrado.

Tabla 2

Esquema de modos de desarrollo de software

<i>Modo de desarrollo</i>	<i>Requisitos</i>	<i>Tamaño</i>	<i>Complejidad</i>	<i>Personas</i>	<i>Experiencia</i>
Orgánico	Poco rígidos	Pequeño (<50KLDC)	Pequeña	Pocas	Mucha
Semiacoplado	Poco/medio	Medio (50 a 300KLDC)	Medio	Medio	Medio
Empotrado	Alto	Grande (>300KLDC)	Alta	Alta	Poca

Nota: ([Boehm, 1981](#)).

El caso del tamaño, se consideran las líneas de código fuente del software en unidades de miles de líneas de código (KLDC, por sus siglas en inglés).

Estimación Proyectos de Software

Método COCOMO

Estos valores constantes, codificados aquí como “a”, “b”, “c” y “d”, son propuestos por el modelo COCOMO para complementar las ecuaciones de cálculo usadas en el modelo.

Valores constantes por modo de desarrollo

<i>Modo de desarrollo</i>	COCOMO Básico <i>a</i>	COCOMO Intermedio <i>A</i>	<i>b</i>	<i>c</i>	<i>d</i>
Orgánico	2.4	3.2	1.05	2.50	0.38
Semiacoplado	3.0		1.12		0.35
Empotrado	3.6	2.8	1.20		0.32

Nota: ([Boehm, 1981](#)).

Estimación Proyectos de Software

Ecuaciones

Estas ecuaciones se utilizan para calcular el esfuerzo nominal en personas/mes (E), tiempo estimado en meses (T) y personal requerido (P).

Ecuaciones por tipo de modelo COCOMO: Básico e intermedio

Ecuación	Submodelo básico	Submodelo intermedio
Esfuerzo (E)	$(E) = a * (KLDC)^b$	$(E) = a * (KLDC)^b * ME$
Tiempo (T)	$(T) = c * (E)^d$	$(T) = c * (E)^d$
Personal (P)	$(P) = E/T$	$(P) = E/T$

Nota: (Boehm, 1981).

Estimación Proyectos de Software

Modelo COCOMO Básico

	ORGANICO	SEMIDESCONECTADO	INTEGRADO
Esfuerzo Estimado [PM]	$E_D = 2,4 \times (KLDC)^{1,05}$	$E_D = 3,0 \times (KLDC)^{1,12}$	$E_D = 3,6 \times (KLDC)^{1,20}$
Tiempo de desarrollo [Meses]	$T_D = 2,5 \times (E_D)^{0,38}$	$T_D = 2,5 \times (E_D)^{0,35}$	$T_D = 2,5 \times (E_D)^{0,32}$
Productividad	$Pr = \frac{LDC}{E_D}$		
Nº medio de personas	$P = \frac{E_D}{T_D}$		
Esfuerzo de mantenimiento [PM]	<p>TCA (Tráfico de Cambio Anual): porción de instrucciones fuente que sufren algún cambio durante un año, bien sea por adición o modificación.</p> $E_M = TCA \cdot E_D$ <p>Y por lo tanto, el número medio de personas a tiempo completo dedicadas a mantenimiento durante 12 meses sería:</p> $P_M = \frac{E_M}{12}$		

Estimación Proyectos de Software

Modelo COCOMO personal requerido

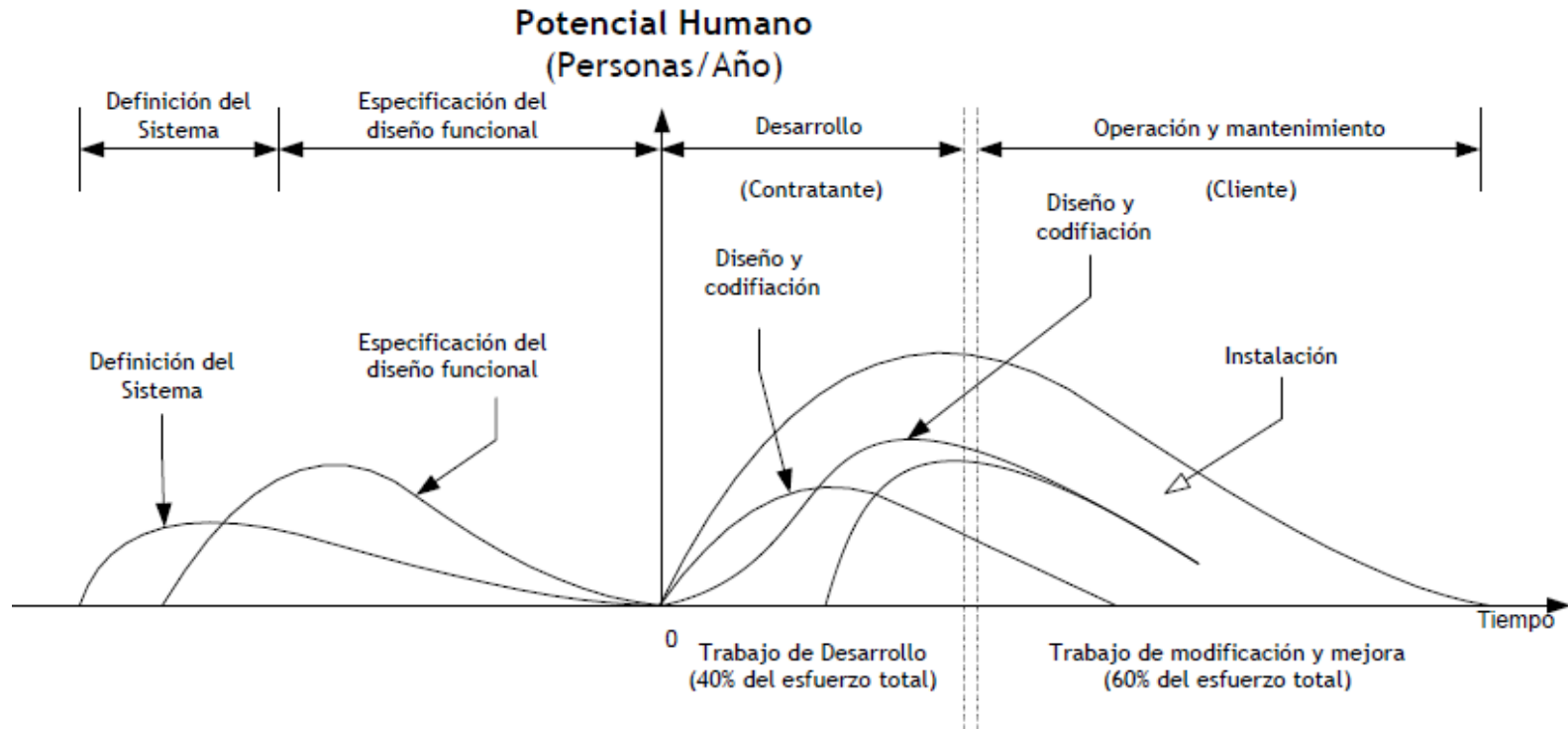


Figura 5.1: Curva de Rayleigh-Norden de esfuerzo de desarrollo

Estimación Proyectos de Software

Modelo COCOMO personal requerido

Ecuaciones por tipo de modelo COCOMO: Básico e intermedio

Multiplicadores de esfuerzo (ME)			Valoración					
			Muy bajo	Bajo	Nominal	Alto	Muy alto	Extr. alto
Atributos del producto								
1.	RELY	Fiabilidad requerida del software	0,75	0,88	1.00	1,15	1,40	
2.	DATA	Tamaño de la base de datos		0,94	1.00	1,08	1,16	
3.	CPLX	Complejidad del producto	0,70	0,85	1.00	1,15	1,30	1,65
Atributos de la computadora								
4.	TIME	Restricciones del tiempo de ejecución			1.00	1,11	1,30	1,66
5.	STOR	Restricciones del almacenamiento princ.			1.00	1,06	1,21	1,56
6.	VIRT	Inestabilidad de la máquina virtual		0,87	1.00	1,15	1,30	
7.	TURN	Tiempo de respuesta del computador		0,87	1.00	1,07	1,15	

Estimación Proyectos de Software

Modelo COCOMO personal requerido

Ecuaciones por tipo de modelo COCOMO: Básico e intermedio

Multiplicadores de esfuerzo (ME)			Valoración				
			Muy bajo	Bajo	Nominal	Alto	Muy alto
Atributos del personal							
8.	ACAP	Capacidad del analista	1,46	1,19	1.00	0,86	0,71
9.	AEXP	Experiencia en la aplicación	1,29	1,13	1.00	0,91	0,82
10.	PCAP	Capacidad de los programadores	1,42	1,17	1.00	0,86	0,70
11.	VEXP	Experiencia en S.O. utilizado	1,21	1,10	1.00	0,90	
12.	LEXP	Experiencia en el lenguaje de progr.	1,14	1,07	1.00	0,95	
Atributos del proyecto							
13.	MODP	Uso de prácticas de programación modernas	1,24	1,10	1.00	0,91	0,82
14.	TOOL	Uso de herramientas software	1,24	1,10	1.00	0,91	0,83
15.	SCED	Restricciones en la duración del proy.	1,23	1,08	1.00	1,04	1,10

Estimación Proyectos de Software

Modelo COCOMO personal requerido

En el primer ejemplo, se desea estimar el costo del proyecto, de forma básica, utilizando el modelo COCOMO básico. El desarrollo de este ejemplo se muestra en la Tabla 6.

Tabla 6

Solución ejemplo #1. COCOMO básico modo semi-acoplado

Ecuación	Submodelo básico	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)^b$	$= 3 * (100)^{1.12}$ $= 3 * 173.78$ $= 521,34$
Tiempo (T)	$(T) = c * (E)^d$	$= 2.5 * (521.34)^{0.35}$ $= 2.5 * 8,93$ $= 22,33$
Personal (P)	$(P) = E/T$	$= 521.34 / 22.3$ $= 23.34$

Estimación Proyectos de Software

Modelo COCOMO Ejemplos

En el primer ejemplo, se desea estimar el costo del proyecto, de forma básica, utilizando el modelo COCOMO básico. El desarrollo de este ejemplo se muestra en la Tabla 6.

Tabla 6

Solución ejemplo #1. COCOMO básico modo semi-acoplado

Ecuación	Submodelo básico	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)^b$	$= 3 * (100)^{1.12}$ $= 3 * 173.78$ $= 521,34$
Tiempo (T)	$(T) = c * (E)^d$	$= 2.5 * (521.34)^{0.35}$ $= 2.5 * 8,93$ $= 22,33$
Personal (P)	$(P) = E/T$	$= 521.34 / 22.3$ $= 23.34$

Estimación Proyectos de Software

Modelo COCOMO Ejemplos ..

En el segundo ejemplo, se desea estimar el costo del proyecto, utilizando el modelo COCOMO intermedio, ajustado con la característica de la *fiabilidad requerida del software* (RELY) con una valoración de “alto”. El desarrollo de este ejemplo se muestra en la Tabla 7.

Tabla 7

Solución ejemplo #2. COCOMO intermedio, modo semiacoplado con ME = RELY “alto”

Ecuación	Submodelo intermedio	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)^b * ME$	$= 3 * (100)^{1.12} * 1.15$ $= 3 * 173.78 * 1.15$ $= 599.54$
Tiempo (T)	$(T) = c * (E)^d$	$= 2.5 * (599.541)^{0.35}$ $= 2.5 * 9.38$ $= 23.45$
Personal (P)	$(P) = E/T$	$= 599.541 / 23.45$ $= 25.57$

Estimación Proyectos de Software

Modelo COCOMO Ejemplos.

En el tercer ejemplo, se desea estimar del costo del proyecto, utilizando el modelo COCOMO intermedio, ajustado con las características *fiabilidad requerida del software (RELY)* con una valoración de “alto”, y *experiencia en la aplicación (AEXP)* con una valoración de “bajo”. El desarrollo de este ejemplo se muestra en la Tabla 8.

Tabla 8

Solución ejemplo #3. COCOMO intermedio, modo semiacoplado los ME RELY “alto” y AEXP “bajo”

Ecuación	Submodelo intermedio	Aplicación de la ecuación
Esfuerzo (E)	$(E) = a * (KLDC)^b * ME$	$= 3 * (100)^{1.12} * 1.15 * 1.13$ $= 3 * 173.78 * 1.299$ $= 677.22$
Tiempo (T)	$(T) = c * (E)^d$	$= 2.5 * (677.22)^{0.35}$ $= 2.5 * 9.78$ $= 24.45$
Personal (P)	$(P) = E/T$	$= 677.22 / 24.45$ $= 27.69$

Fuente:

ISSN Electrónico: 2215-3470 DOI:
<http://dx.doi.org/10.15359/ru.32-1.8>
Gabriela Garita-González y Fulvio
Lizano-Madriz



Universidad
Israel

Gracias

Responsabilidad con pensamiento positivo

Tareas



Universidad
Israel

- **Aplicar la estimación de costos punto función a su proyecto de clase**
- **Aplicar método COCOMO al proyecto utilizando herramienta COCOMOII**

NOTA: El deber será enviado en formato PDF, especificando su nombre y número de semana y número grupo

Ejemplo: **Semana1_Mario_Pérez_Grupo1.pdf**