

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

GESTION DE PROYECTOS DE SOFTWARE

**TEMA: Estimación de Proyectos de
Software**

Ing. Jorge Geovanny Raura Ruiz Phd
jgraura@espe.edu.ec

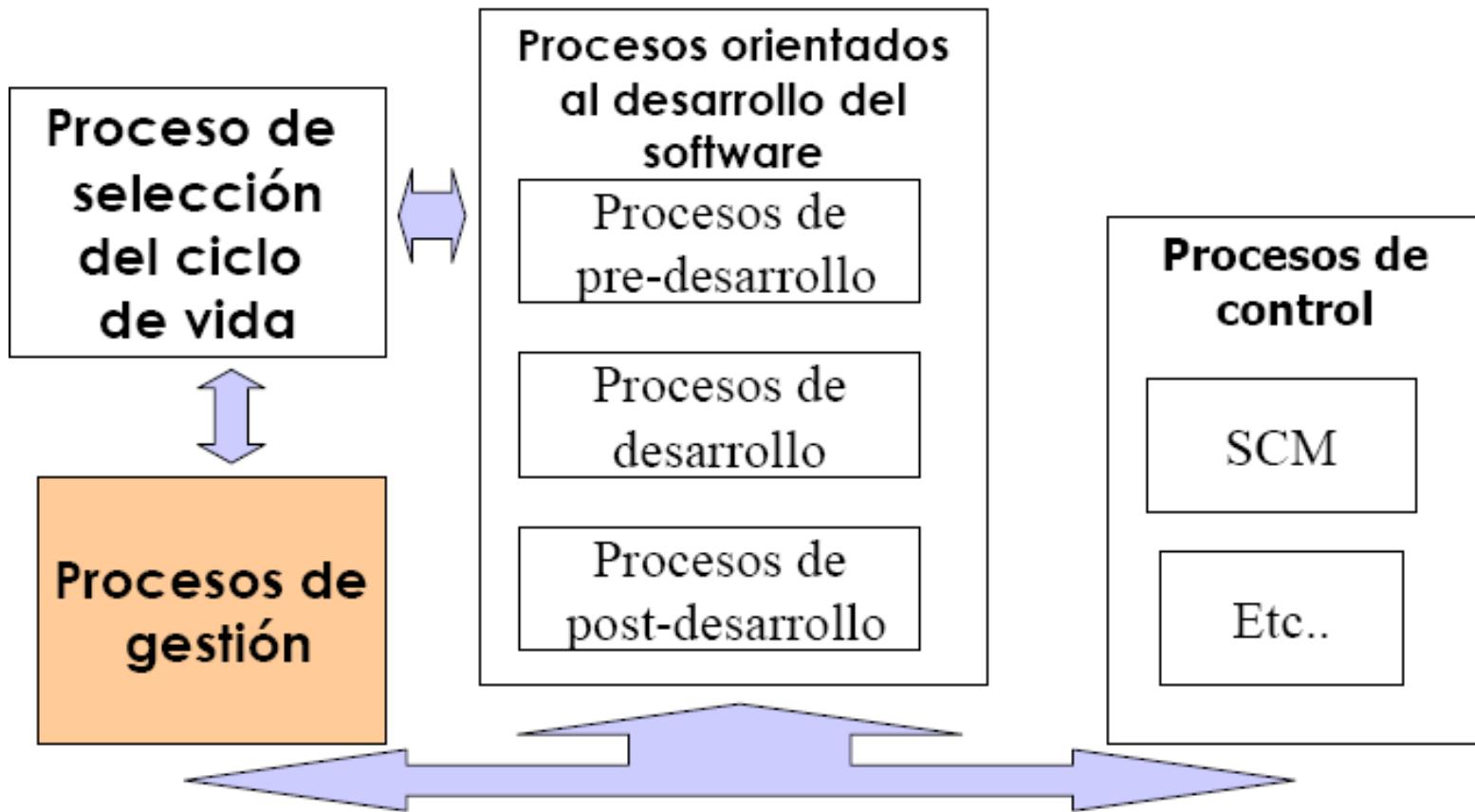
CONTENIDO

- 1. Introducción**
- 2. Gestión de Proyectos**
- 3. El Proceso de Estimación**
- 4. Modelos de Estimación**
- 5. La Realidad del Desarrollo de Software en el Ecuador**

1. INTRODUCCION

“Es necesario construir en un **tiempo corto**, sin un **costo excesivo**, aplicaciones de software complejas, de **calidad** y que soporten las necesidades del usuario. Estas aplicaciones deberían ser fáciles y rápidas de **modificar**”.

2. GESTION DE PROYECTOS – IEEE 1074



GESTION DEL PROYECTO

IEEE 1074

ACTIVIDADES DEL
PROCESO DE GESTIÓN DEL
PROYECTO

INICIACIÓN DEL
PROYECTO

PLANIFICACIÓN
DEL PROYECTO

SEGUIMIENTO
Y CONTROL DEL
PROYECTO

GESTION DE PROYECTOS

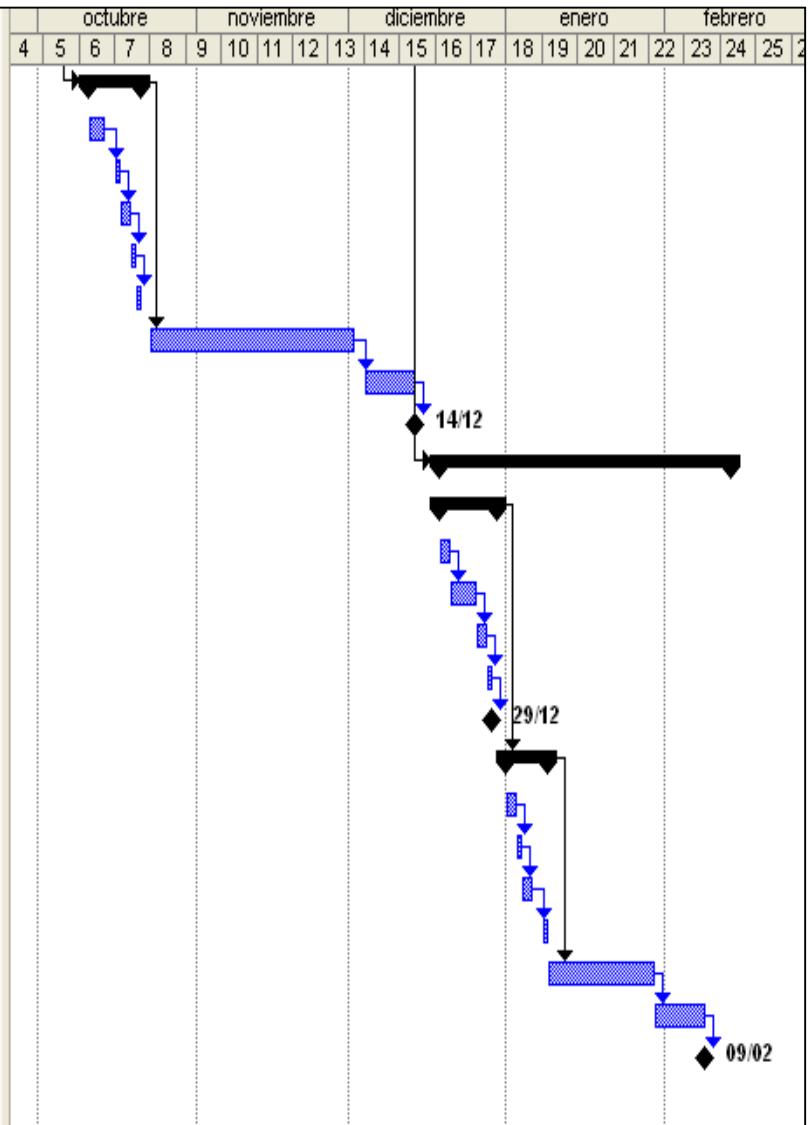
- Los proyectos son un conjunto de actividades y recursos ordenados en el tiempo y que desarrollan un producto, bien o servicio.
- La gestión se centra en la coordinación de las actividades y recursos en el tiempo.
- Las técnicas de gestión existentes no están orientadas a la gestión específica de proyectos de software pero pueden adaptarse.

LA PLANIFICACION

- Se define como: “**la predicción de la duración de las actividades** y tareas a nivel individual, los recursos requeridos, la concurrencia y solapamiento de tareas para ser desarrollados en paralelo y el camino crítico a través de la red de actividades”

LA PLANIFICACION

	Nombre de tarea	Duración	Comienzo	Fin	Predec
19	■ Diseño	8 días	mié 11/10/06	vie 20/10/06	13
20	Diagramas de secuencia	3 días	mié 11/10/06	vie 13/10/06	
21	Modelo de clases de diseño	1 día	lun 16/10/06	lun 16/10/06	20
22	Modelo de datos	2 días	mar 17/10/06	mié 18/10/06	21
23	Modelo interface	1 día	jue 19/10/06	jue 19/10/06	22
24	Revisión LB Diseño	1 día	vie 20/10/06	vie 20/10/06	23
25	Implementación	30 días	lun 23/10/06	vie 01/12/06	19
26	Pruebas	8 días	lun 04/12/06	mié 13/12/06	25
27	Revisión LB Ciclo 1	3 días	jue 14/12/06	lun 18/12/06	26
28	■ Segundo ciclo de desarrollo	41 días	mar 19/12/06	mar 13/02/07	12
29	■ Análisis	9 días	mar 19/12/06	vie 29/12/06	
30	Caso de uso expandidos	2 días	mar 19/12/06	mié 20/12/06	
31	Diagramas de secuencia del sistema	3 días	jue 21/12/06	lun 25/12/06	30
32	Contratos	2 días	mar 26/12/06	mié 27/12/06	31
33	Modelo conceptual	1 día	jue 28/12/06	jue 28/12/06	32
34	Revisión LB Análisis	1 día	vie 29/12/06	vie 29/12/06	33
35	■ Diseño	6 días	lun 01/01/07	lun 08/01/07	29
36	Diagramas de secuencia	2 días	lun 01/01/07	mar 02/01/07	
37	Modelo de clases de diseño ciclo 2	1 día	mié 03/01/07	mié 03/01/07	36
38	Revisión Modelo de datos	2 días	jue 04/01/07	vie 05/01/07	37
39	Revisión LB Diseño	1 día	lun 08/01/07	lun 08/01/07	38
40	Implementación	15 días	mer 09/01/07	lun 29/01/07	35
41	Pruebas	8 días	mar 30/01/07	jue 08/02/07	40
42	Revisión LB Ciclo 2	3 días	vie 09/02/07	mar 13/02/07	41



LA ESTIMACIÓN

- Se define como: “la predicción de personal, esfuerzo, y costos que se requerirá para terminar todas las actividades y productos conocidos asociados con el proyecto”. A. Moreno.
- Una “actividad que permite obtener principalmente respuestas aproximadas a las siguientes preguntas: Cuánto cuesta?, Cuánto tiempo llevará hacerlo?” Larry Putman.

POR QUÉ ES DIFÍCIL ESTIMAR?

- No existe un modelo de estimación universal.
- Hay muchas personas implicadas en los proyectos que precisan estimaciones para toma de decisiones.
- La utilidad de una estimación también dependerá de la etapa de desarrollo nos encontremos.

POR QUÉ ES DIFÍCIL ESTIMAR?

- La estimación a menudo se hace superficialmente.
- Las estimaciones claras, completas son difíciles de formular.
- Las características del software y de su desarrollo hacen difícil la estimación.
- La tecnología y metodologías cambian con rapidez lo que implica que el proceso de estimación no se estabilicen.
- Un estimador puede no tener experiencia en estimar desarrollos.

POR QUÉ ES DIFÍCIL ESTIMAR?

- Un estimador puede no tener experiencia en estimar desarrollos.
- Existe una tendencia aparente de los desarrolladores hacia la subestimación.
- El estimador infiere en base a su experiencia tiempos de desarrollo. No toma en cuenta que el trabajo puede ser hecho por personal menos experimentado.
- Malas interpretaciones en la relación personal-tiempo.
- Tendencia a reducir las estimaciones para mejorar la oferta.
- Los factores de esfuerzo y tamaño (drivers de costo) en general son difíciles de determinar.

Discusión

- Qué requisitos debe cumplir un buen estimador?
- Cuándo se debe realizar la estimación?
- Qué es lo que debemos estimar? (además del tiempo y costo)

3. EL PROCESO DE ESTIMACION

CASO PRACTICO

- Un Disk Jockey desea tener un pequeño sistema de información.

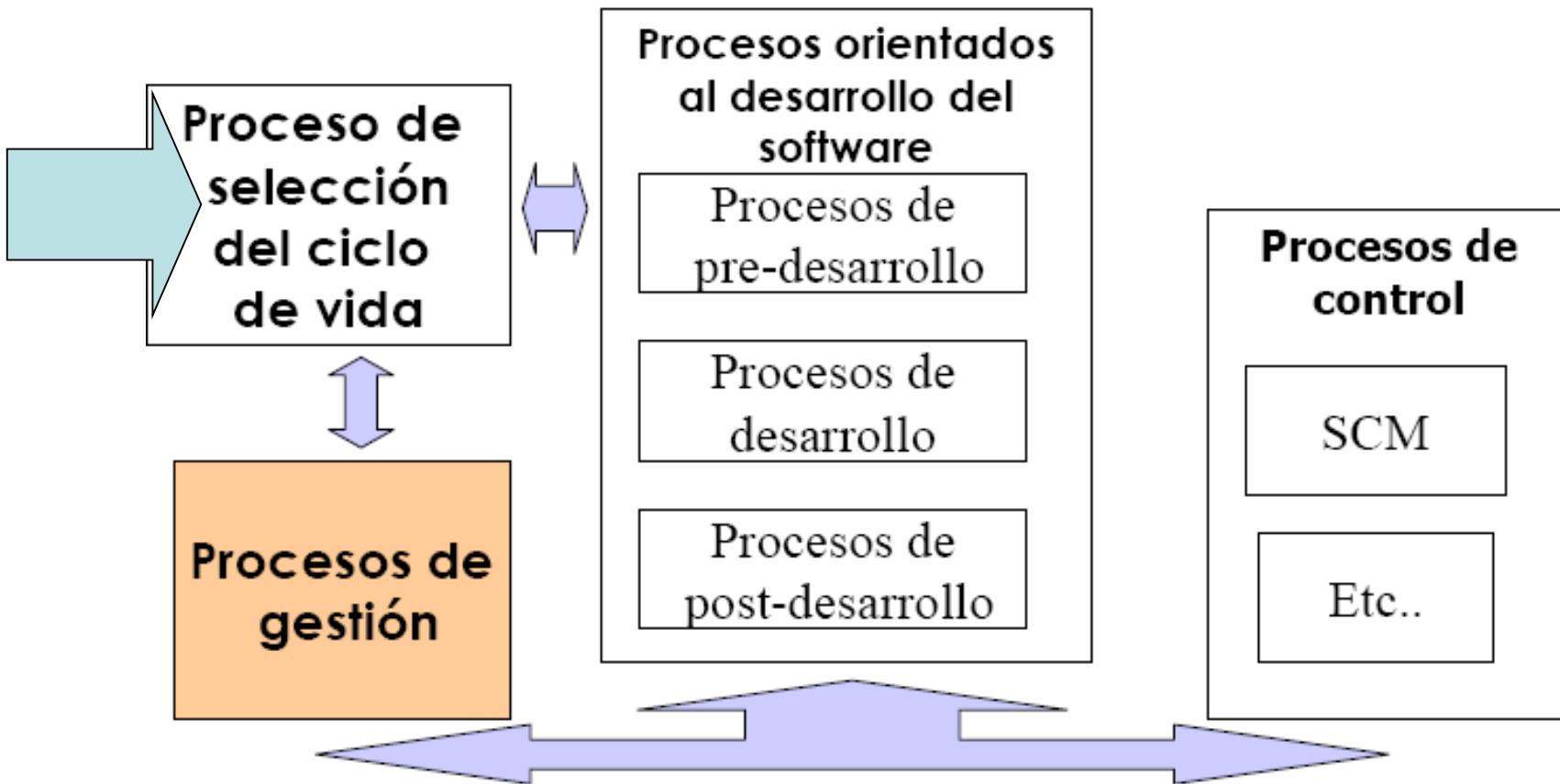


ESPECIFICACION DE REQUISITOS

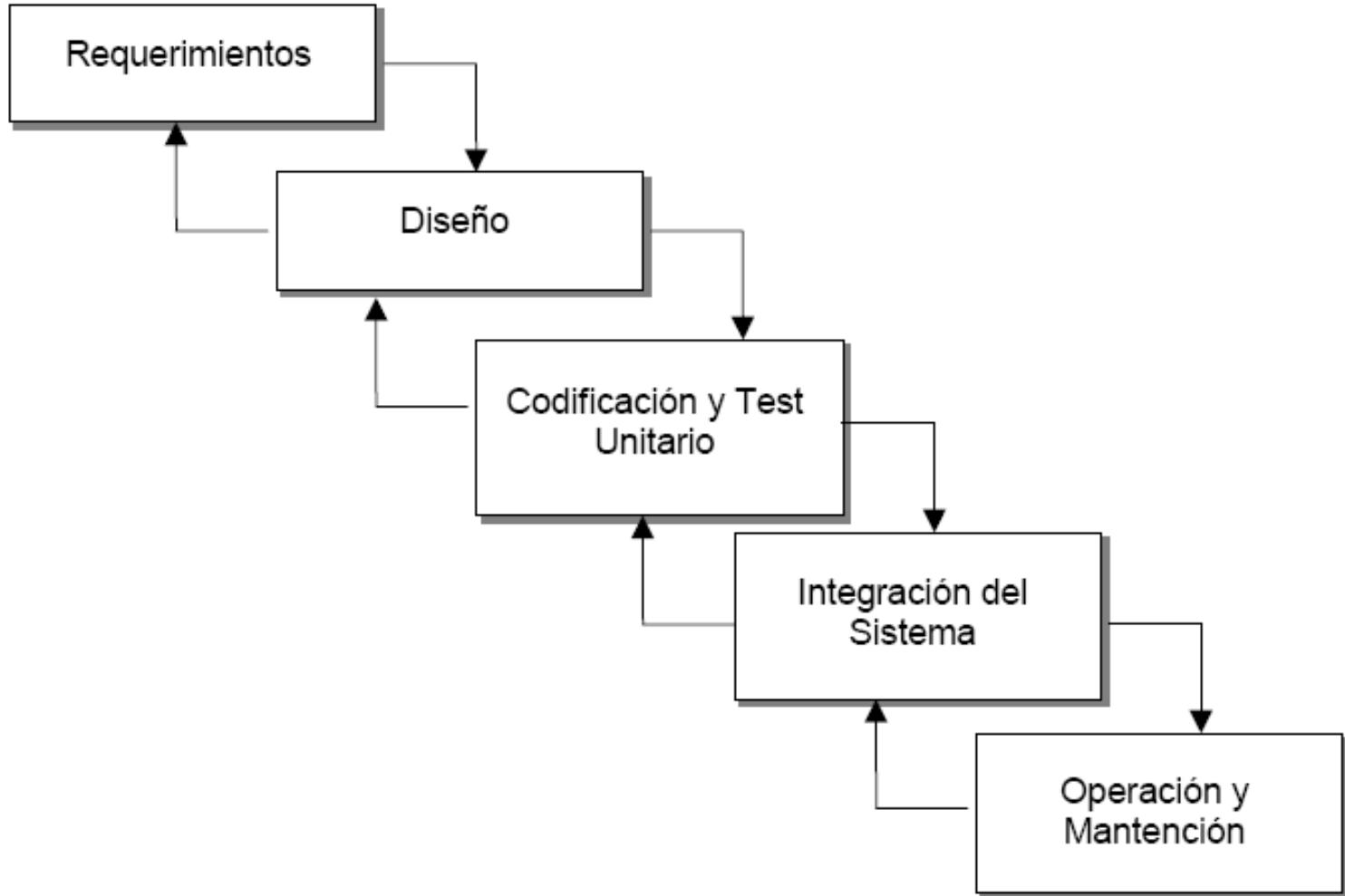
- **R1:** El sistema permitirá la administración (Ingreso, Modificación, eliminación) de CDs.
- **R2:** Un CD contiene la siguiente Información:
 - Título del CD.
 - Grupo Musical
 - Productor
 - Fecha de Producción
 - Título de la Canción
 - Nombre del Cantante
 - Autor de la Canción
 - Tiempo de duración
- **R3:** El sistema permitirá consultar los datos del CD por su Título.
- **R4:** El sistema generará un reporte del total de CD por grupo musical.
- **R5:** Se generará un reporte sobre el tiempo de reverberación tomando como coeficiente de absorción el promedio de los materiales del local y asumiendo un volumen cúbico.
- Existen otros requisitos que el cliente aún no los tiene claros en esta etapa.

GESTION DE PROYECTOS –

IEEE 1074

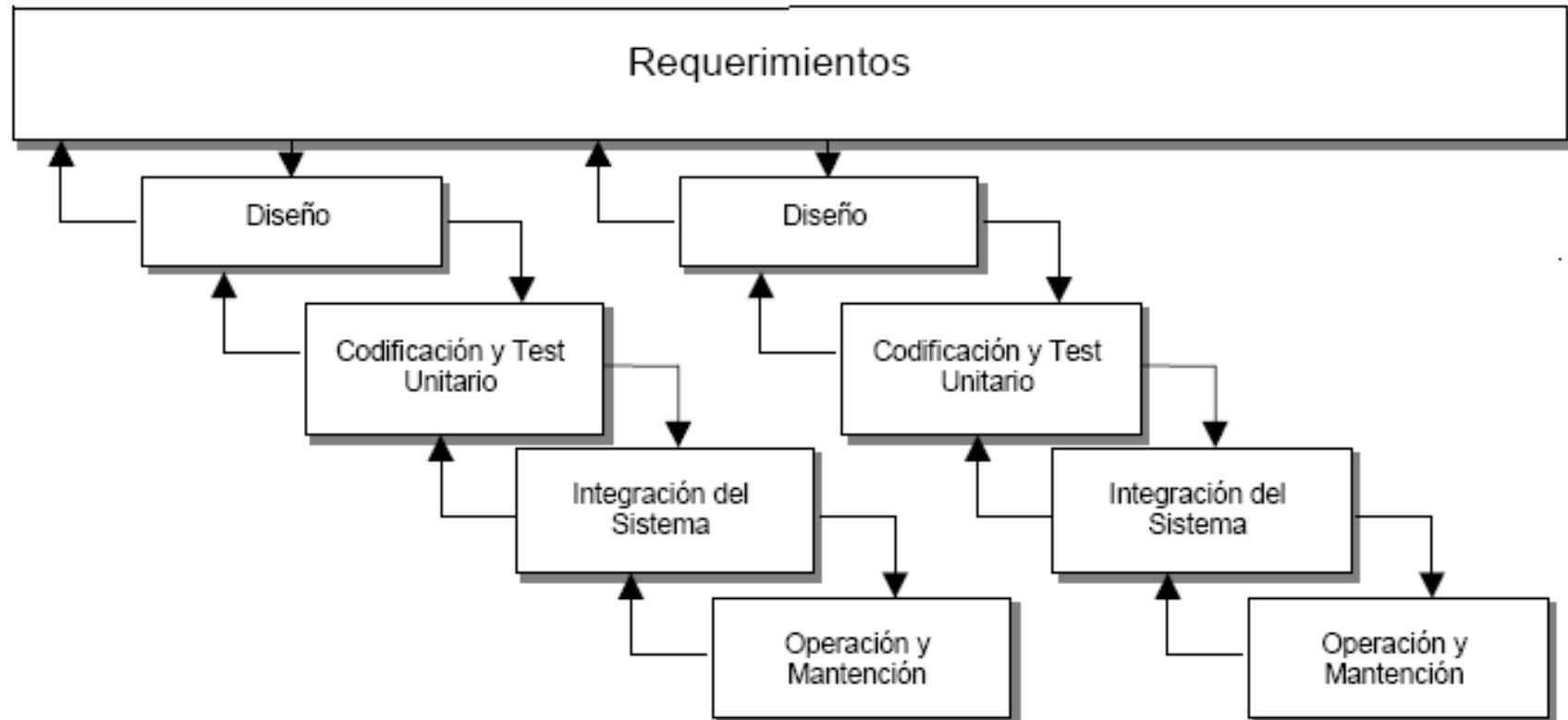


SELECCIÓN DEL CICLO DE VIDA



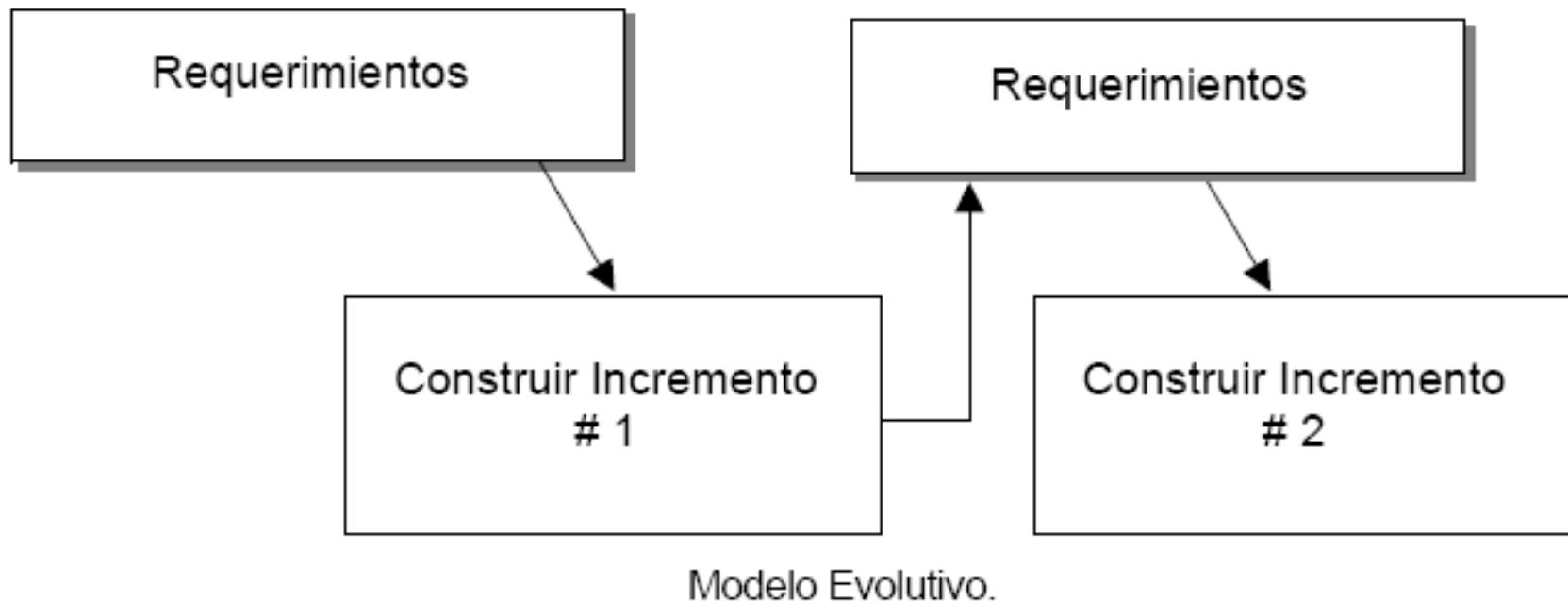
Modelo de Ciclo de Vida Cascada.

SELECCIÓN DEL CICLO DE VIDA

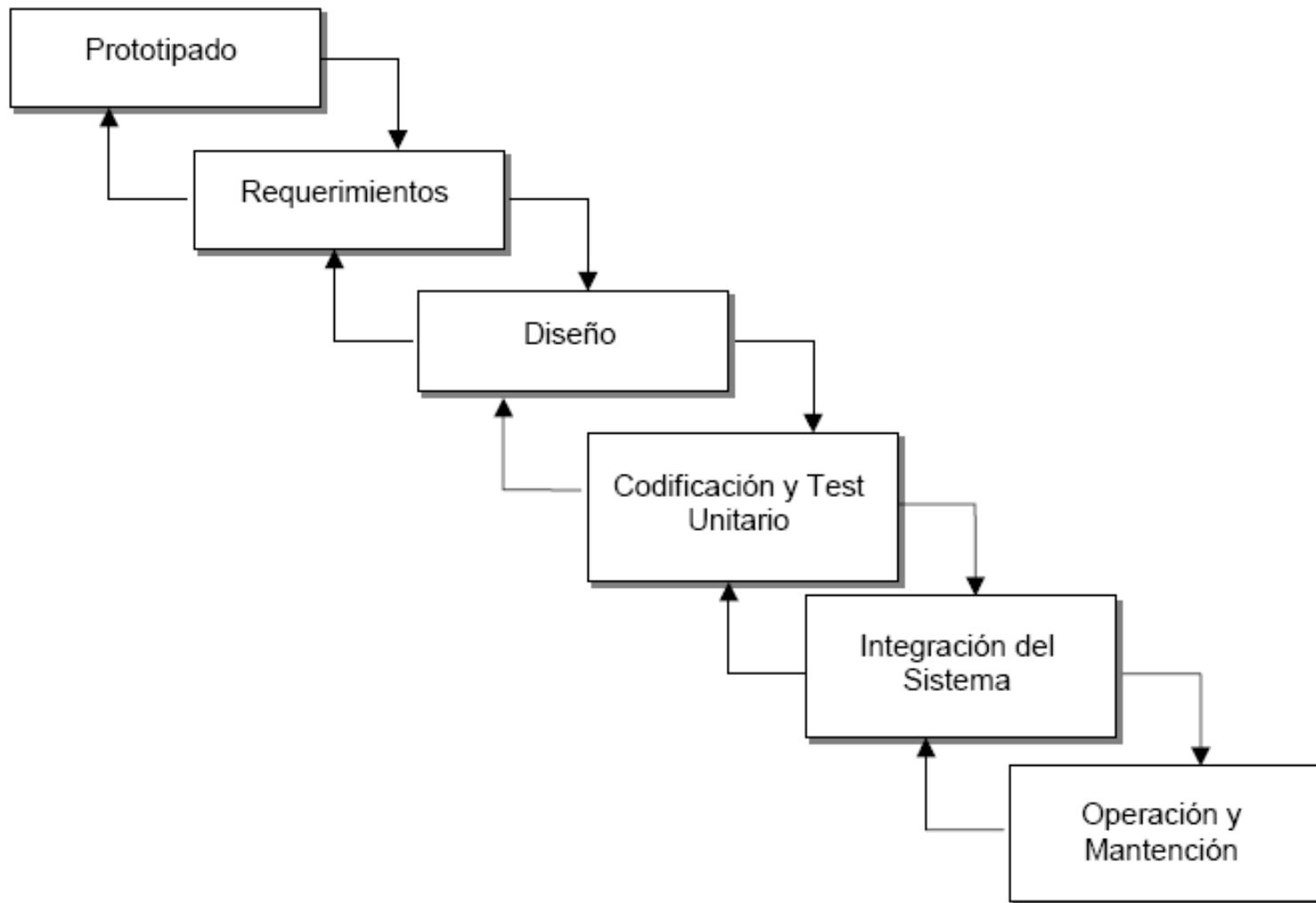


Modelo de Desarrollo Incremental con desarrollo en cascada de los incrementos.

SELECCIÓN DEL CICLO DE VIDA

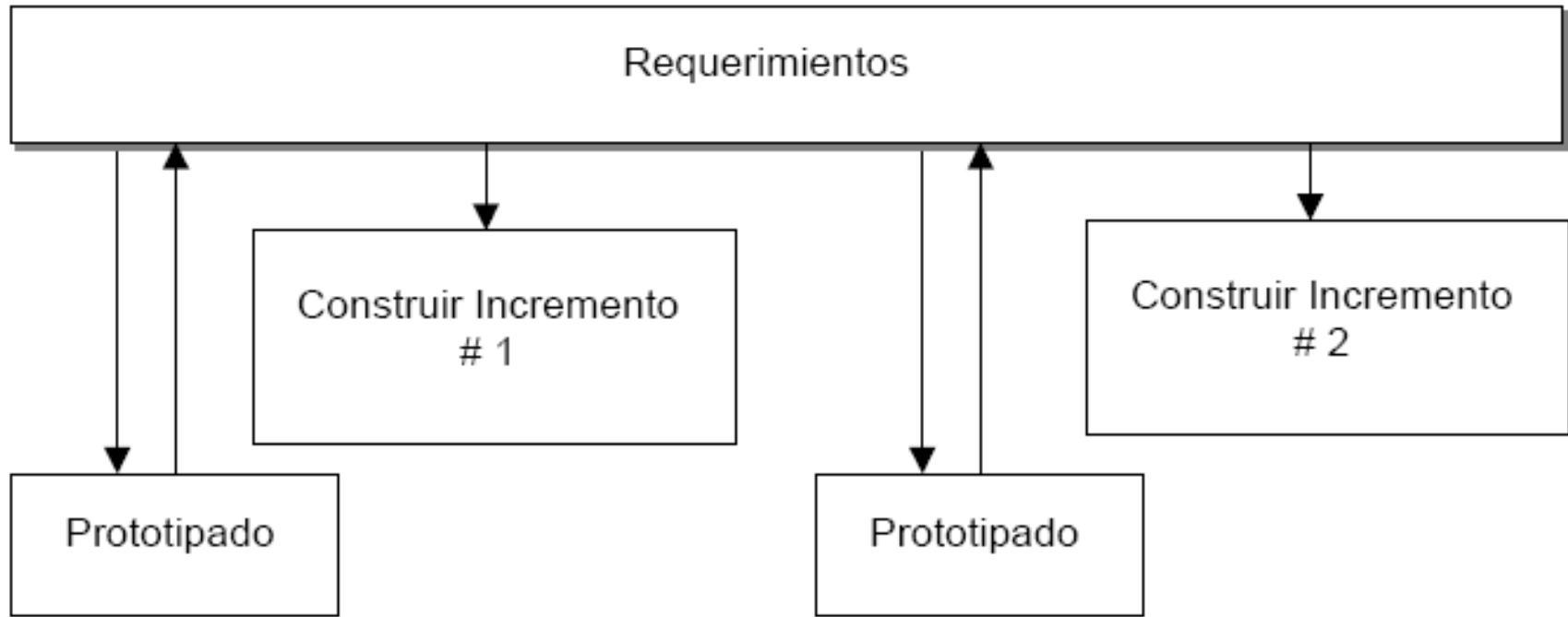


SELECCIÓN DEL CICLO DE VIDA



Prototipado de Requerimientos basado en el modelo Cascada.

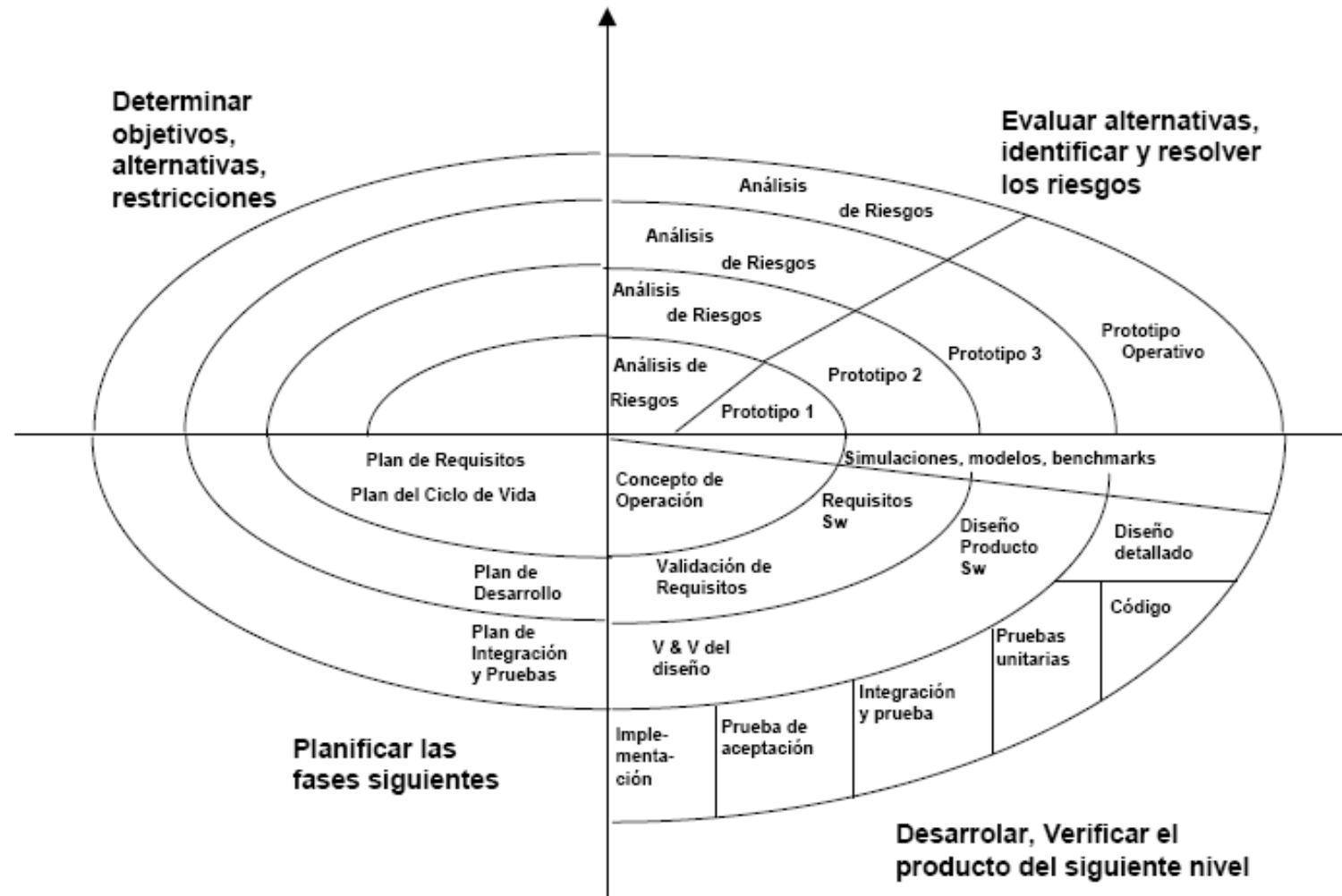
SELECCIÓN DEL CICLO DE VIDA



Prototipado de Requerimientos basado en el modelo de desarrollo incremental.

SELECCIÓN DEL CICLO DE VIDA

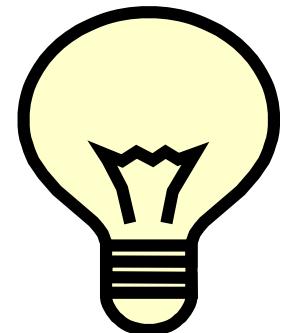
MODELO EN ESPIRAL



SELECCIÓN DEL CICLO DE VIDA

Criterios a considerar:

- Madurez de la aplicación (relacionado a la probabilidad que muchos requerimientos comenzarán a conocerse sólo después del uso del sistema).
- Complejidad del problema y de la solución.
- Frecuencias y magnitudes esperadas de los cambios de requerimientos.
- Financiamiento disponible, y su perfil como una función del tiempo.
- Acceso de los desarrolladores a los usuarios.
- Certeza de requerimientos conocidos.



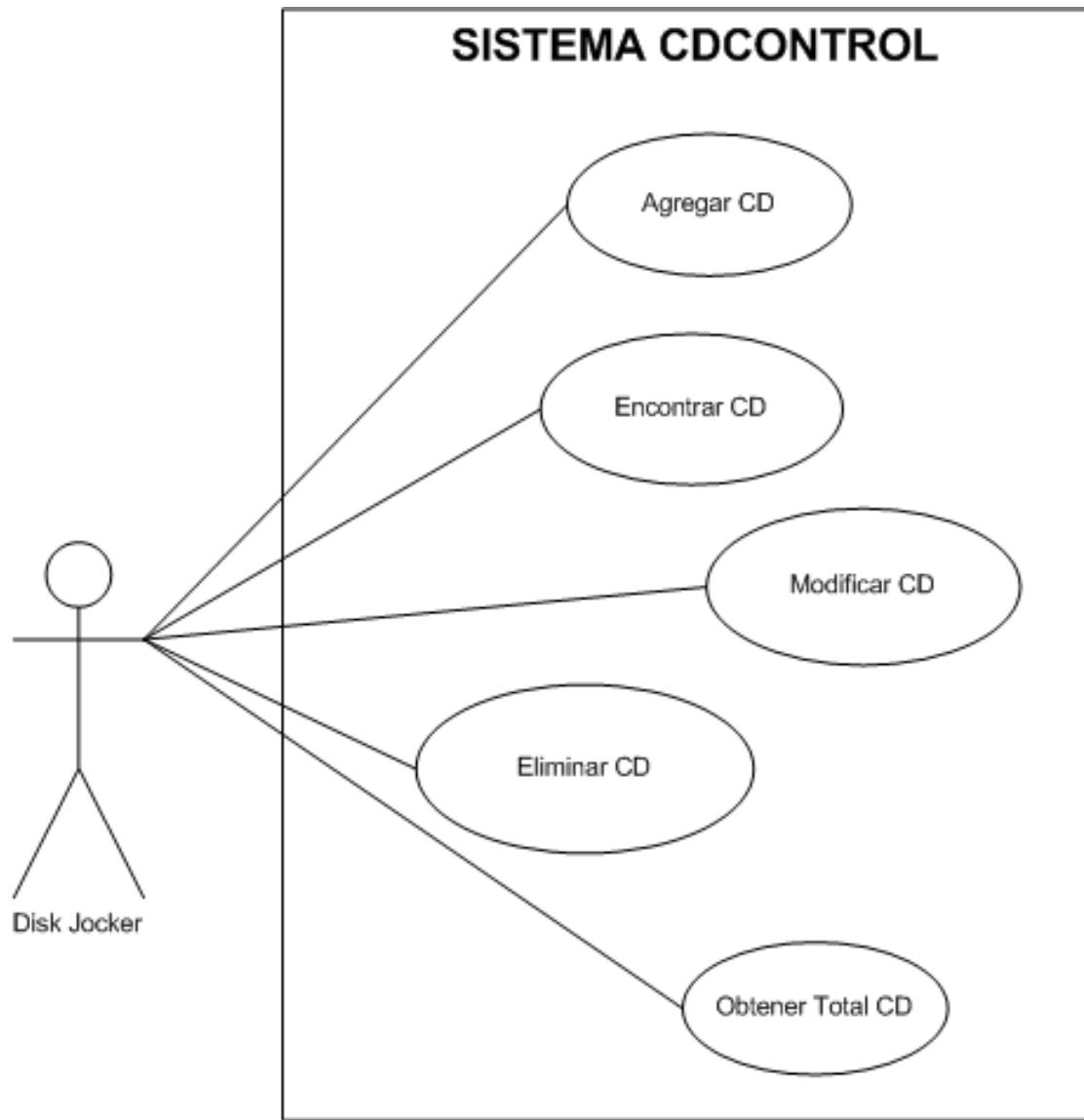
CRITERIOS DE SELECCIÓN DEL CICLO DE VIDA

- Para el caso de estudio se ha seleccionado el ciclo de vida evolutivo basado en el desarrollo de un prototipo:
 - No existe certeza en el conocimiento de todos los requisitos (Ver Req 5). Se justifica la realización de un PROTOTIPO.
 - No se conocen bien todos los requisitos. (aplicación no madura)
 - Es decir presencia de INCERTIDUMBRE.



MODELO DE CASOS DE USO

Primera Iteración (Incremento)



DEFINICION DE CASOS DE USO DE ALTO NIVEL

CASO DE USO 1

<i>Caso de Uso</i>	AGREGAR CD	
<i>Actor</i>	Disk Jockey	
<i>Tipo</i>	Primario y esencial	
<i>Propósito</i>	Registrar un CD en el Sistema.	
<i>Visión General:</i>	Permite que el Disk Jockey efectúe el alta de un CD en el sistema.	
<i>Referencias:</i>	Req 1, 2	
<i>Curso Típico De Eventos</i>	<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
	1. El Disk Jockey ingresa los datos del CD: Título del CD, Grupo Musical Productor, Fecha de Producción, Título de la Canción, Nombre del Cantante, Autor de la Canción, Tiempo de duración	2. Incorpora la información del CD en su base de datos,

CASOS DE USO 2

<i>Caso de Uso</i>	ENCONTRAR CD	
<i>Actor</i>	Disk Jockey	
<i>Tipo</i>	Primario y esencial	
<i>Propósito</i>	Localizar un CD en el Sistema.	
<i>Visión General:</i>	Permite que el Disk Jockey encuentre un CD en el sistema.	
<i>Referencias:</i>	Req 3	
<i>Curso Típico De Eventos</i>	<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
	1. El usuario ingresa el Título del CD.	2. Ubica el CD y muestra el Título del CD. Grupo Musical, Título de la Canción, Nombre del Cantante, Autor de la Canción, Tiempo de duración

CASO DE USO 3

<i>Caso de Uso</i>	MODIFICAR CD
<i>Actor</i>	Disk Jockey
<i>Tipo</i>	Primario y esencial
<i>Propósito</i>	Registrar un CD en el Sistema.
<i>Visión General:</i>	Permite que el Disk Jockey modifique la información de CD en el sistema.
<i>Referencias:</i>	Req 1, 2

<i>Curso Típico De Eventos</i>	<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
	<ol style="list-style-type: none">1. El usuario localiza el CD e ingresa los datos que desea modificar (Título del CD, Grupo Musical Productor, Fecha de Producción, Título de la Canción, Nombre del Cantante, Autor de la Canción, Tiempo de duración)	<ol style="list-style-type: none">2. Modifica la información del CD en su base de datos.

CASO DE USO 4

<i>Caso de Uso</i>	ELIMINAR CD	
<i>Actor</i>	Disk Jockey	
<i>Tipo</i>	Primario y esencial	
<i>Propósito</i>	Eliminar un CD en el Sistema.	
<i>Visión General:</i>	Permite que el Disk Jockey elimine un CD del sistema.	
<i>Referencias:</i>	<i>Req 1,2</i>	
<i>Curso Típico De Eventos</i>	<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
	1. El usuario localiza el CD a eliminar y confirma el borrado del mismo	2. Elimina la información del CD en su base de datos,

CASO DE USO 5

<i>Caso de Uso</i>	OBTENER TOTAL CD	
<i>Actor</i>	Disk Jockey	
<i>Tipo</i>	Primario y esencial	
<i>Propósito</i>	Obtener un reporte del total de CD registrados en el Sistema.	
<i>Visión General:</i>	Permite mostrar un reporte con el total de CD producidos por un grupo musical	
<i>Referencias:</i>	Req 3	
<i>Curso Típico De Eventos</i>	<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
	1. El usuario pide al sistema el total de CD registrados.	2. Muestra un reporte con el nombre del Grupo Musical y el total de CD por grupo.

PREGUNTAS INMEDIATAS DEL CLIENTE

- Cuánto cuesta?
- En qué tiempo me entrega?

-

Es bastante si estas preguntas llegan luego de tener por lo menos una primera ERS.



DISCUSIÓN: PREGUNTAS INMEDIATAS DEL DESARROLLADOR



- Trabajar con la Intuición. (ojímetro) ?
- Basarme en mi experiencia – analogía (si la tengo).?
- Descomponer el proyecto (divide y vencerás) ?
- Basarme en datos históricos (si realicé mediciones). ?
- Preguntar a un experto. (tipo Delphi)?
- Utilizar un modelo de estimación (???).

Algo más que se nos ocurra?

ES NECESARIO DECIDIR

- Vamos a suponer dos escenarios:
 - 1) Tengo datos históricos. (Un buen caso)
 - 2) No tengo experiencia, no hay expertos en el tema, no tengo datos históricos, no quiero ser adivino ... (El peor caso)



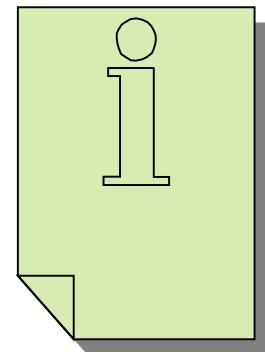
ESCENARIO 1

- Tengo Datos Históricos:

Proyecto (Se asume dos personas trabajando)	Miles de líneas de código VB (KLOC) x	Tiempo de Desarrollo (Meses) y
Soft A	1	1
Soft B	2	1
Soft C	3	2
Soft D	4	2
Soft E	5	4

DEMOSTRAR:

1. Se puede general el siguiente modelo de estimación empírica:
 $\hat{y} = \beta_0 + \beta_1 x = -0.1 + 0.7x$
2. $H_0 : \beta_1 = 0 ; H_a : \beta_1 \neq 0$ con $\alpha = 0.05$, si $t_{.025} = 3.182$
3. El Intervalo de confianza de 95% es [0.09 a 1.31]
4. El Coeficiente de correlación $r = 0.904$; con $r[-1, 0, +1]$
5. El coeficiente de determinación $r^2 = 0.82$



Es decir:

El tiempo de desarrollo aumenta en 0.7 meses por cada 1000 Líneas de código que se escriban. Si asumimos que la aplicación tendrá en promedio unas 7KSLOC, entonces el tiempo de desarrollo será $t = 4.8$ Meses?

Si por el contrario, No se desarrolla ninguna línea de código, ($x=0$) tiene sentido un tiempo negativo ? ($\hat{y} = -0.1$?)

Cómo mejorar el modelo (disminuir el intervalo de confianza) ?

ESCENARIO 2

- No tengo experiencia, no hay expertos en el tema, no tengo datos históricos, no quiero ser adivino ...

La única alternativa: Qué modelos de estimación existen? (COCOMO, SOFTCOST, SPQR, Jensen/SEER, ESTIMACS...)



MODELOS DE ESTIMACIÓN

Dominados por ciclos de vida en cascada – 60's – 80's

Prominent software estimation models

Model	Year
SDC (Systems Development Corp.)	1965
TRW Wolverton	1974
Putnam SLIM (<i>Software Lifecycle Management</i>)	1976
Boeing Black	1977
Doty	1977
IBM-FSD (IBM Federal Systems Division)	1977
RCA Price S	1977
Walston-Felix	1977
IBM function points	1979
Bailey-Basili Meta-Model	1981
Cocomo (Constructive Cost Model)	1981
SoftCost-R	1981
Estimacs	1983
Jensen/SEER (Software Evaluation and Estimation of Resources)	1983
SPQR (Software Productivity Quality and Reliability)/Checkpoint	1985

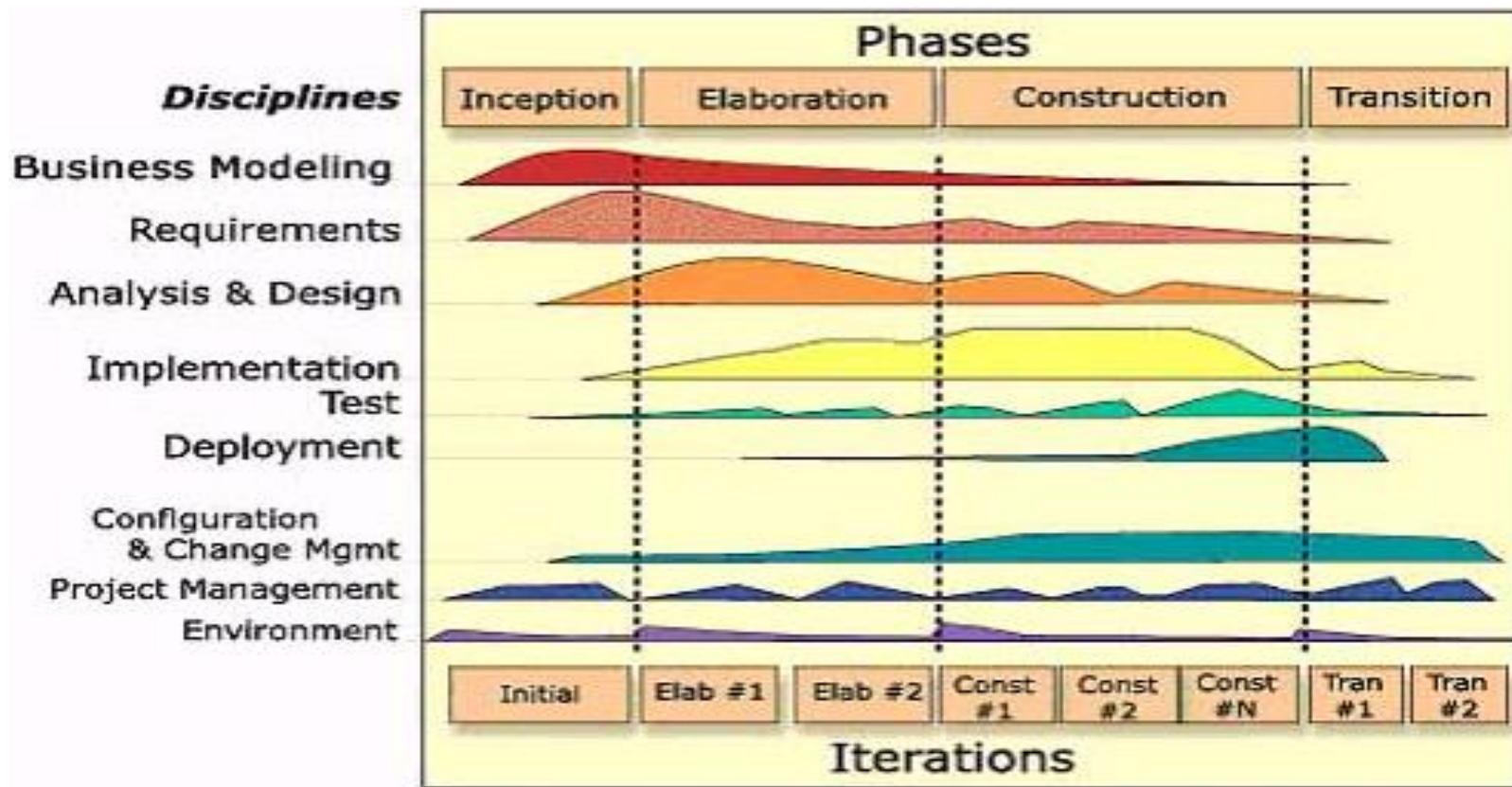
4. COCOMO

- Acrónimo de COnstructive COst MOdel.
- Es el modelo empírico más utilizado en la industria.
- Tiene algunos años de continuo desarrollo, propuesto por Barry Boehm en 1.981 (COCOMO 81). Ultima versión COCOMO II 2000.
- Permite la estimación de esfuerzo, tiempo y costo de los proyectos de desarrollo de software.

COCOMO

- La estimación cubre las etapas de Elaboración y Construcción propuestos por RUP/MBASE.
- Incluye todos los costos directos del proyecto, pero no los indirectos.
- Los datos empíricos que soportan el modelo se obtienen de una **muestra limitada de proyectos**. (83 proyectos en la versión inicial COCOMO II)

COCOMO - ETAPAS



Inception Elaboration Construction Transition

MODELOS DE ESTIMACION COCOMO II

- El modelo de Composición de aplicaciones.
- El modelo de Diseño anticipado.
- El modelo Post-Arquitectura.

El modelo de Composición de Aplicaciones

- Indicado para proyectos construidos con herramientas modernas de construcción de interfaces gráficos para usuario.
- Meta: Programación de Usuario Final.
- Medida: Puntos Objeto

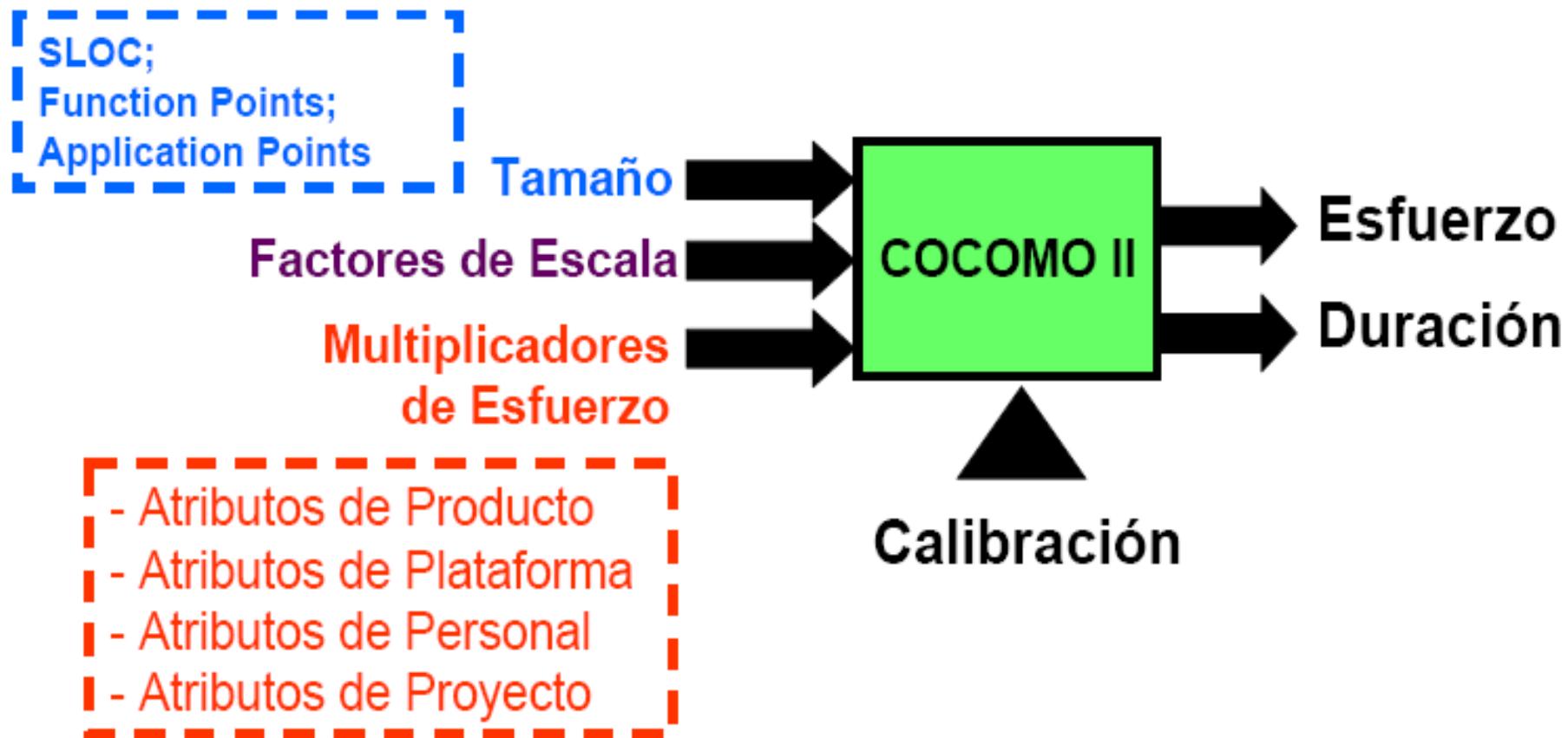
El modelo de Diseño anticipado

- Este modelo puede utilizarse para obtener estimaciones aproximadas del costo de un proyecto antes de que esté determinada por completo su arquitectura.
- Utiliza un pequeño conjunto de drivers de costo (7).
- Medida: Puntos de Función sin ajustar o KSLOC (Miles de Líneas de Código Fuente)(depende del lenguaje - Backfiring).

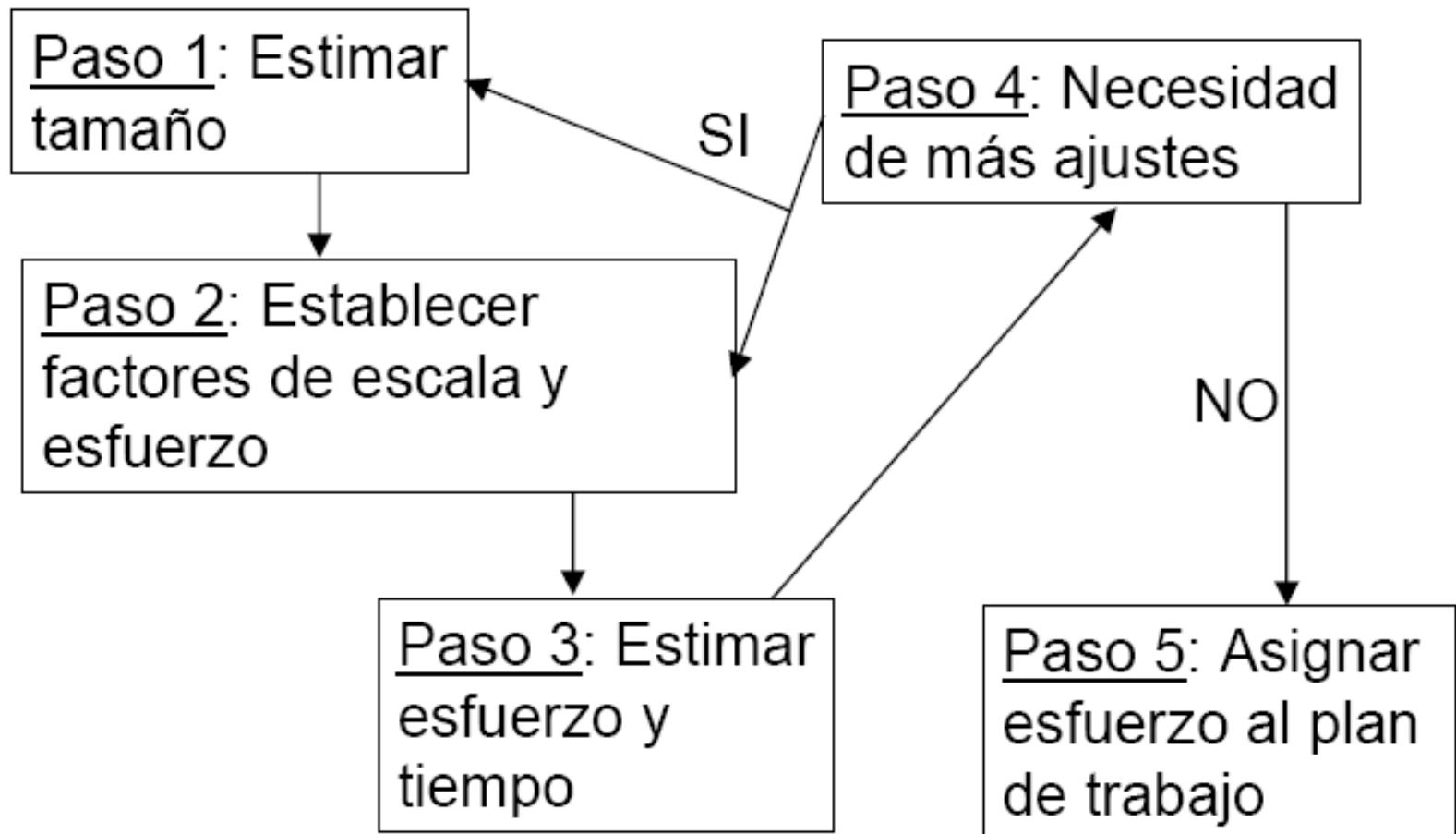
El modelo Post-Arquitectura

- Es el modelo COCOMO II más detallado.
- Se utiliza una vez que se ha desarrollado por completo la arquitectura del proyecto.
- Modela en base a 17 drivers de costo.
- Medida: Puntos de Función sin ajustar o KSLOC (Miles de Líneas de Código Fuente)

CONCEPTO OPERACIONAL COCOMO II



Proceso de estimación con COCOMO II



ECUACIÓN DE CÁLCULO DE ESFUERZO

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

Donde:

PM = Esfuerzo medido en Personas-Mes

A = constante tomada por defecto del modelo,
ajustado en 2.94

Size = tamaño del software (KSLOC)

B = factores de escala

EL TAMAÑO IMPORTA?



CÓMO MEDIR EL TAMAÑO DEL SOFTWARE?

- Líneas de Código (LOC).
 - Es factible si se tiene código desarrollado.
 - Su definición no es clara y depende del lenguaje.
- Puntos de Función (PF)
 - Permite medir la funcionalidad del sistema desde la perspectiva del usuario.
 - Existen algunos métodos de conteo (Albrecht IFPUG, MKII, NESMA, COSMIC-FFP.)

ES NECESARIO DECIDIR

- Se utilizará el método de conteo de puntos de función propuesto por Albrecht.
- Es un estándar de la industria (ISO/IEC 20926:2003 Software engineering).
- Actualmente es el más conocido y utilizado.
- Se ajusta a las características del caso de estudio.



MÉTRICA DE PUNTOS DE FUNCIÓN DE ALBRECHT

Métrica de los Puntos de Función

- Es una métrica aceptada como estándar en el mercado.
 - IFPUG (International Function Point Users Group).
 - CPM 4.0 de 1994 (Counting Practice Manual)
- Inicialmente Albrecht en IBM. (1979)
- ¿Como medir el software?

Métrica de los Puntos de Función

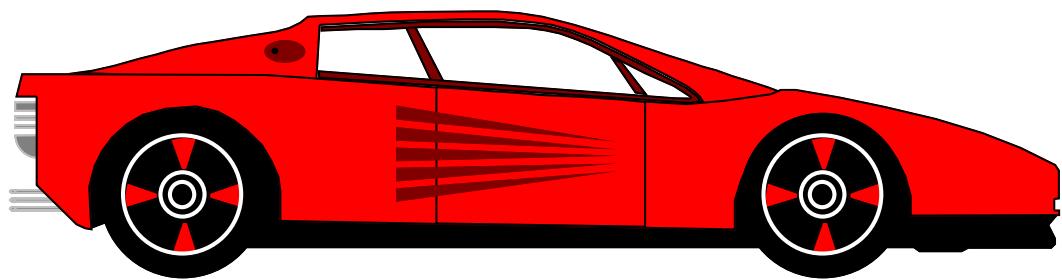
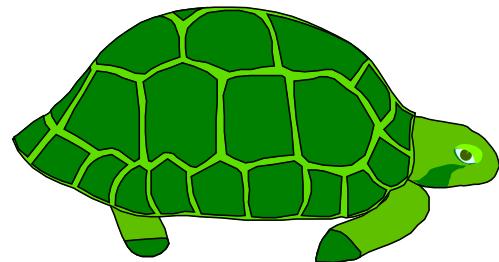
- Es una métrica que se puede aplicar en las primeras fases de desarrollo.
- Se basa en características fundamentalmente “Externas” de la aplicación a desarrollar.
- Mide dos tipos de características:
 - Los elementos de función (entradas, salidas, ficheros, etc.)
 - Los factores de Complejidad.

Elementos de Función

- Son elementos fácilmente identificables en los diagramas de especificación del sistema. (DFD, Entidad-Relación, DD)
- Los usuarios los entienden perfectamente.
- Observamos la aplicación como una caja negra.

Visión de caja negra

- Nos centramos en características visibles del objeto en estudio.
- Ejemplo:
 - Equipo de música.
 - Auto
 - Animales



Elementos de función

- Entradas – EI - P:186
- Salidas - EO - P:223
- Consultas – EQ – P: 257
- Ficheros Lógicos o Internos – ILF – P: 68
- Ficheros de Interfaz Externa – EIF – P: 108

Algunas Definiciones

- Proceso elemental
- Datos e información de control
- Lógica de proceso
 - Ediciones, algoritmos o cálculos
 - Accesos a ficheros para consulta o actualización

Proceso elemental:

- Menor unidad de actividad que tiene sentido para el usuario, conocedor del sistema en estudio.

Datos e informaciones de control:

- Datos elementales con los que trabaja la aplicación en estudio. Nos referimos a ellos siempre como datos aunque se componen de los datos propios del sistema en estudio, más las informaciones de control que solicita el usuario: mensajes de error, claves de seguridad... etc.

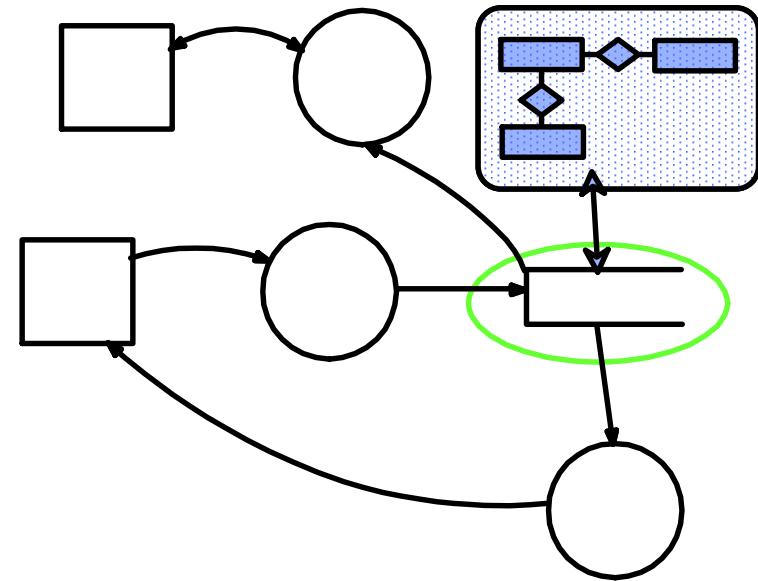
Lógica de proceso:

- Procesos que se producen como consecuencia de un proceso elemental.
Pueden ser de dos tipos:
 - Ediciones, algoritmos o cálculos
 - Accesos a un fichero para mantenimiento (consulta, eliminación o actualización)

Ficheros Lógicos o Internos

ILF

- Agrupaciones de datos, tal y **como los percibe el usuario**
- Es diferente de:
 - Entidades y Relaciones
 - Tablas o archivos resultantes del diseño físico
- Los grupos de datos serán accedidos y actualizados por la aplicación



REGLAS DE IDENTIFICACIÓN DE FICHEROS LÓGICOS INTERNOS - ILF

Cuestión:	Respuesta, debe ser
Se trata de una agrupación de datos lógica o identifiable desde el punto de vista del usuario y satisface un requerimiento específico del usuario	Si
La agrupación de datos es mantenida por procesos de la aplicación en estudio	Si
La agrupación de datos es mantenida mediante un proceso elemental de la aplicación	Si
La agrupación de datos no ha sido contada como un fichero de interfaz externo	Si

Clasificación de los Ficheros Lógicos o Internos

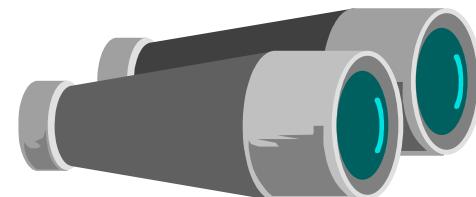
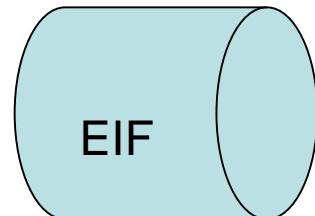
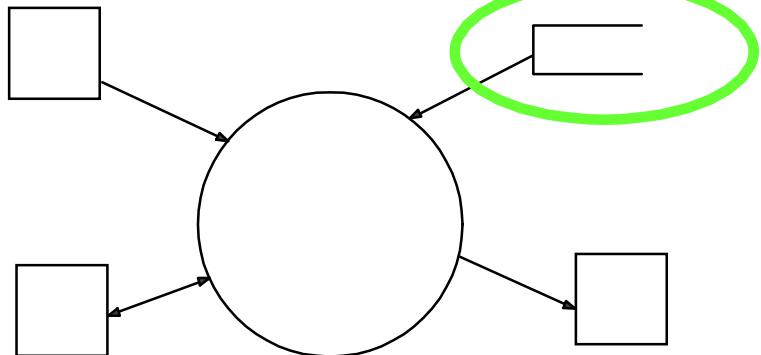
DIFICULTAD FICHEROS	Número de Campos o Atributos DET (Data Elements)		
	1-19 Atributos	20-50 Atributos	51 + Atributos
1 Registro Lógico	BAJA	BAJA	MEDIA
2 a 5 Registros Lógicos	BAJA	MEDIA	ALTA
6 o más Registros Lógic.	MEDIA	ALTA	ALTA

RET (Record Element Types)

Ficheros de Interfaz Externa EIF

- Ficheros a los que accede la aplicación con el único objetivo de **obtener información**.
- Son mantenidos por otras aplicaciones
- Nunca los actualiza la aplicación.

DIAGRAMA DE CONTEXTO



REGLAS PARA IDENTIFICAR FICHEROS DE INTERFAZ EXTERNOS - EIF

Cuestión:	Respuesta, debe ser
Se trata de una agrupación de datos lógica o identifiable desde el punto de vista del usuario y satisface un requerimiento específico del usuario	Si
La agrupación de datos es referenciada, y externa, a la aplicación en estudio	Si
La agrupación de datos no es mantenida mediante la aplicación en estudio	Si
La agrupación de datos ha sido contada como un fichero lógico Interno en otra aplicación	Si
La agrupación de datos no ha sido contada como un fichero lógico Interno de la aplicación en estudio	Si

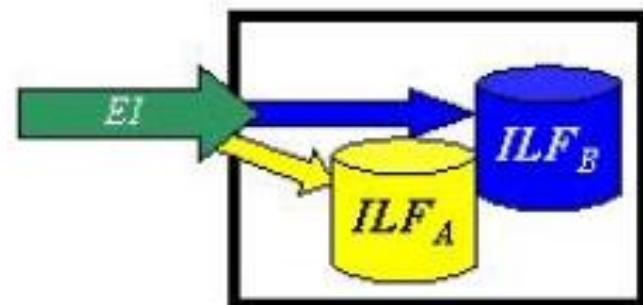
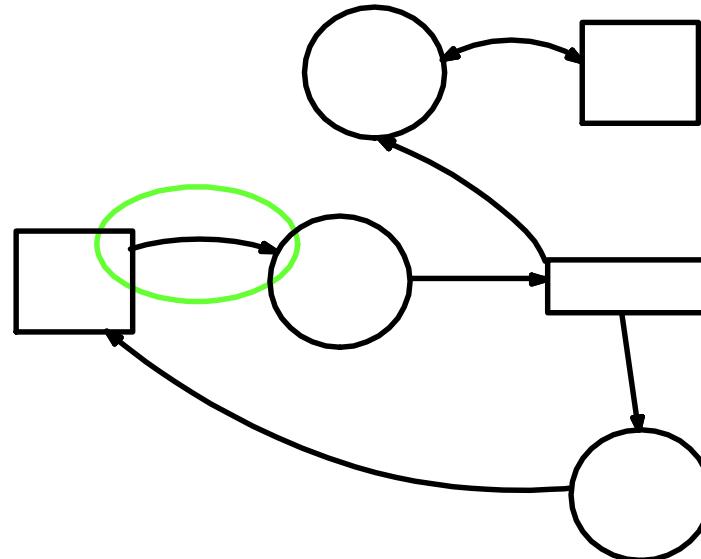
Clasificación de los Ficheros de Interfaz Externos

DIFICULTAD FICHEROS DE INTERFAZ	Número de Campos o Atributos DET (Data Elements)		
	1-19 Atributos	20-50 Atributos	51 + Atributos
1 Entidad o Registro Lógico	BAJA	BAJA	MEDIA
2 a 5 Registros Lógico	BAJA	MEDIA	ALTA
6 o más Registros Lógic.	MEDIA	ALTA	ALTA

RET (Record Element Types)

Entradas Externas EI (External Inputs)

- Informaciones que llegan a la aplicación desde el exterior.
- Tienen una sola dirección (Exterior à Interior)
- Siempre actualizan algún fichero interno.



Reglas para identificar Entradas - EI

Cuestión:	Respuesta, debe ser
Entran datos desde exterior de la aplicación	Si
Existen datos en algún fichero lógico interno que son actualizados	Si
El proceso es la unidad mínima de actividad que tiene sentido para el usuario	Si
El proceso es completo y deja al sistema en un estado consistente	SI
Para el proceso subyacente se debe de cumplir alguna de las siguientes reglas	A o B
A La lógica del proceso es exclusiva de esta entrada, o la primera vez que la contamos	
B Los datos elementales son diferentes de otras entradas	

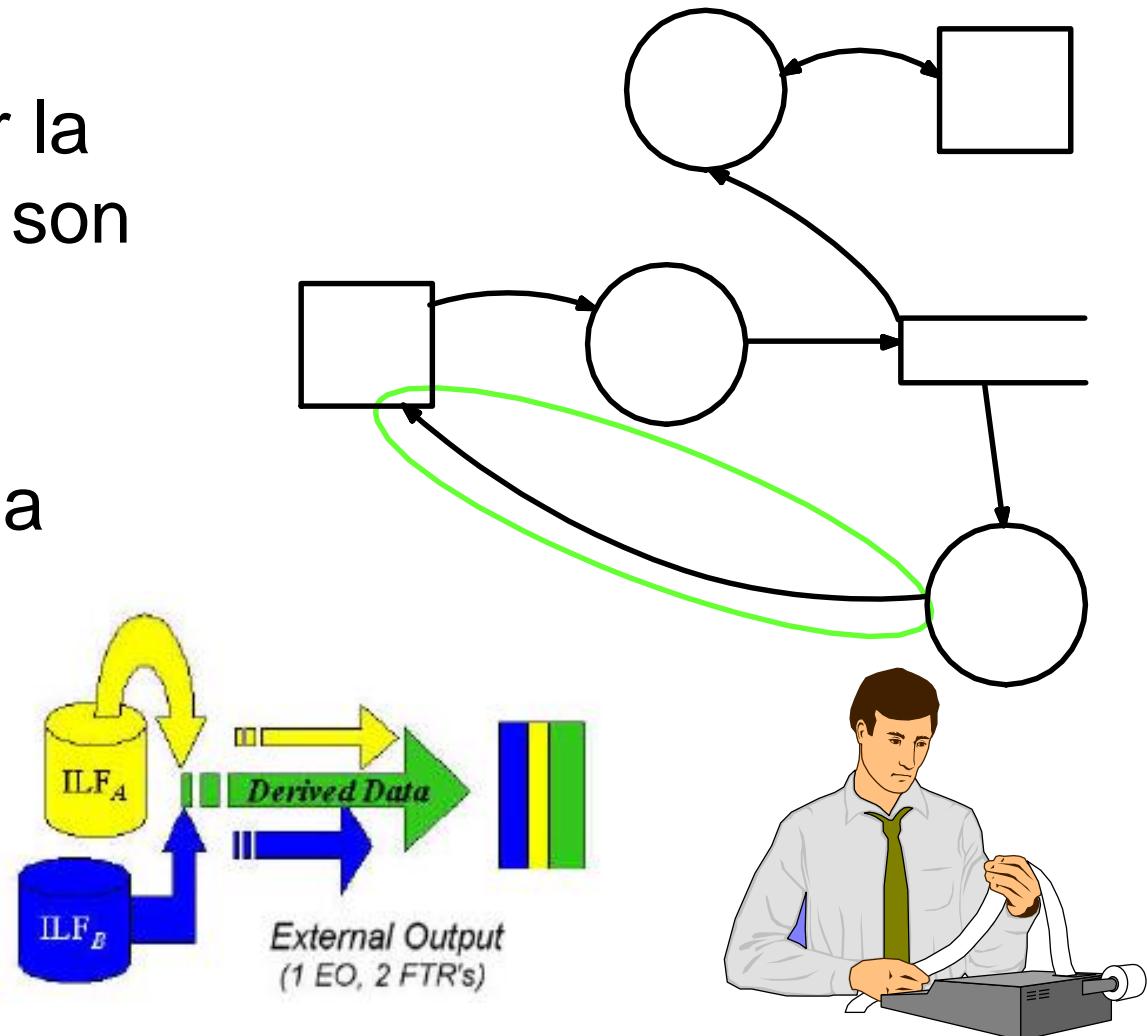
Clasificación de las entradas EI

DIFICULTAD ENTRADAS	Número de Campos o Atributos de la Entrada DET (Data Elements)		
	1-4 Atributos	5-15 Atributos	16 + Atributos
0 ó 1 ficheros accedidos	BAJA	BAJA	MEDIA
2 ficheros accedidos	BAJA	MEDIA	ALTA
3 + ficheros accedidos	MEDIA	ALTA	ALTA

FTR (File Types Referenced)

Salidas Externas – External Outputs EO

- Informaciones elaboradas por la aplicación que son transmitidas al usuario.
- Tienen una sola dirección
(Interior a Exterior)



Reglas para identificar las Salidas Externas - EO

Cuestión:	Respuesta
El proceso envía datos o información al exterior de la aplicación	Si
El proceso es la unidad mínima de actividad que tiene sentido para el usuario	Si
El proceso es completo y deja al sistema en un estado consistente	Si
Para el proceso subyacente se debe de cumplir alguna de las siguientes reglas	A o B
A La lógica del proceso es exclusiva de esta salida (o la primera vez)	
B Los datos elementales son diferentes de otras salidas	

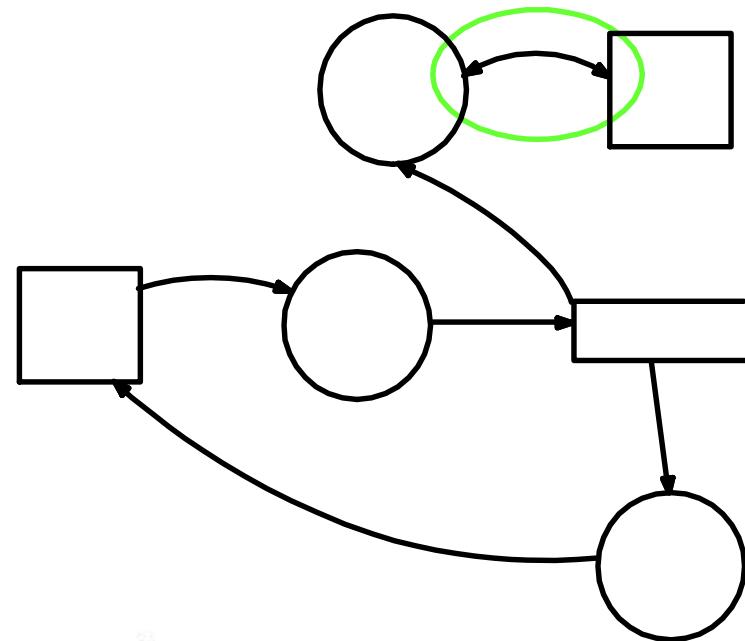
Clasificación de las salidas EO

DIFICULTAD SALIDAS	Número de Campos o Atributos de la Salida DET (Data Elements)		
	1-5 Atributos	6-19 Atributos	20 + Atributos
0 ó 1 ficheros accedidos	BAJA	BAJA	MEDIA
2 ó 3 ficheros accedidos	BAJA	MEDIA	ALTA
4 + ficheros accedidos	MEDIA	ALTA	ALTA

RET (Record Element Types)

Consultas Externas – External Inquiry EQ

- Entradas que producen inmediatamente una salida
- No modifica los datos del sistema



Reglas para identificar Consultas Externas - EQ

Cuestión:	Respuesta,
Una petición atraviesa la frontera del sistema	Sí
El proceso envía datos o información al exterior de la aplicación	Sí
Se recuperan datos	Sí
No se calculan datos derivados para enviar al exterior	Sí
El proceso (entrada/salida) es la unidad mínima de actividad que tiene sentido para el usuario	Sí
El proceso es completo y deja al sistema en un estado consistente	Sí
El proceso no actualiza ningún Fichero Lógico Interno	Sí
Para el proceso subyacente se debe de cumplir alguna de las siguientes reglas	A o B
A La lógica del proceso en su parte de entrada o salida, es distinta del de otras consulta del sistema (o la primera vez)	
B Los datos elementales de la entrada o salida son diferentes de otras consultas	

Clasificación de las consultas EQ

- Calculamos la complejidad de la parte de entrada
- Calculamos la complejidad de la parte de salida
- Nos quedamos sólo con la complejidad mayor de las dos.

Clasificación de las salidas EQ

DIFICULTAD SALIDAS	Número de Campos o Atributos de la Salida DET (Data Elements)		
	1-5 Atributos	6-19 Atributos	20 + Atributos
0 ó 1 ficheros accedidos	BAJA	BAJA	MEDIA
2 ó 3 ficheros accedidos	BAJA	MEDIA	ALTA
4 + ficheros accedidos	MEDIA	ALTA	ALTA

RET (Record Element Types)

Hoja para calcular los Puntos de función sin ajustar (PFSA)

FACTORES DE COMPLEJIDAD

- Son catorce factores que completan la visión externa de la aplicación.
- No están recogidos en la funcionalidad de la aplicación.
- Toman un valor entre 0 y 5

SIGNIFICADO DEL VALOR DE CADA F.C.

Valor	Significado del valor
0	Sin influencia, factor no presente
1	Influencia insignificante, muy baja
2	Influencia moderada o baja
3	Influencia media, normal
4	Influencia alta, significativa
5	Influencia muy alta, esencial

Ejemplo: Comunicación de Datos.

- Los datos usados en el sistema se envían o reciben por líneas de comunicaciones.

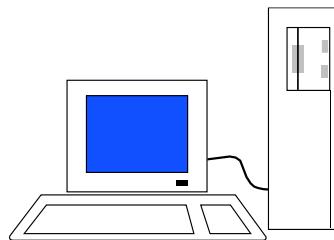
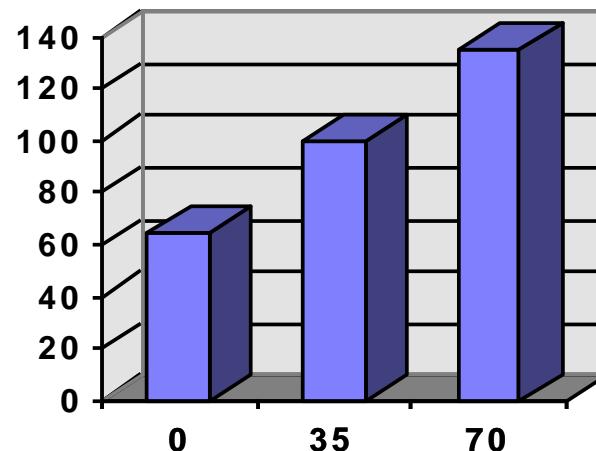


Tabla para el cálculo de los FC.

#	Factor de Complejidad	Valor (0..5)
1	Comunicación de Datos.	
2	Proceso Distribuido.	
3	Rendimiento	
4	Configuración Operacional compartida	
5	Ratio de Transacciones	
6	Entrada de Datos EN-LÍNEA	
7	Eficiencia con el Usuario Final	
8	Actualizaciones EN-LÍNEA	
9	Complejidad del Proceso Interno	
10	Reusabilidad del Código	
11	Contempla la Conversión e Instalación	
12	Facilidad de Operación (back up, etc.)	
13	Instalaciones Múltiples	
14	Facilidad de Cambios	
Factor de Complejidad Total (FCT)		↗ Valor _i

Calculo de los puntos de función ajustados

- $PFA = PFSA * (0,65 + (0.01 * FC))$
- Cada factor de complejidad afecta en +/- 2,5% en los PFSA
- $PFSA * 65\% \leq PFA \leq PFSA * 135\%$



Estimación del Esfuerzo Requerido

- Partimos de los datos históricos de la Organización
- Esfuerzo =
 $PFA * \text{Promedio}_\text{Organización}(\text{Lenguaje})$

Estimación del Esfuerzo Requerido (Datos históricos)

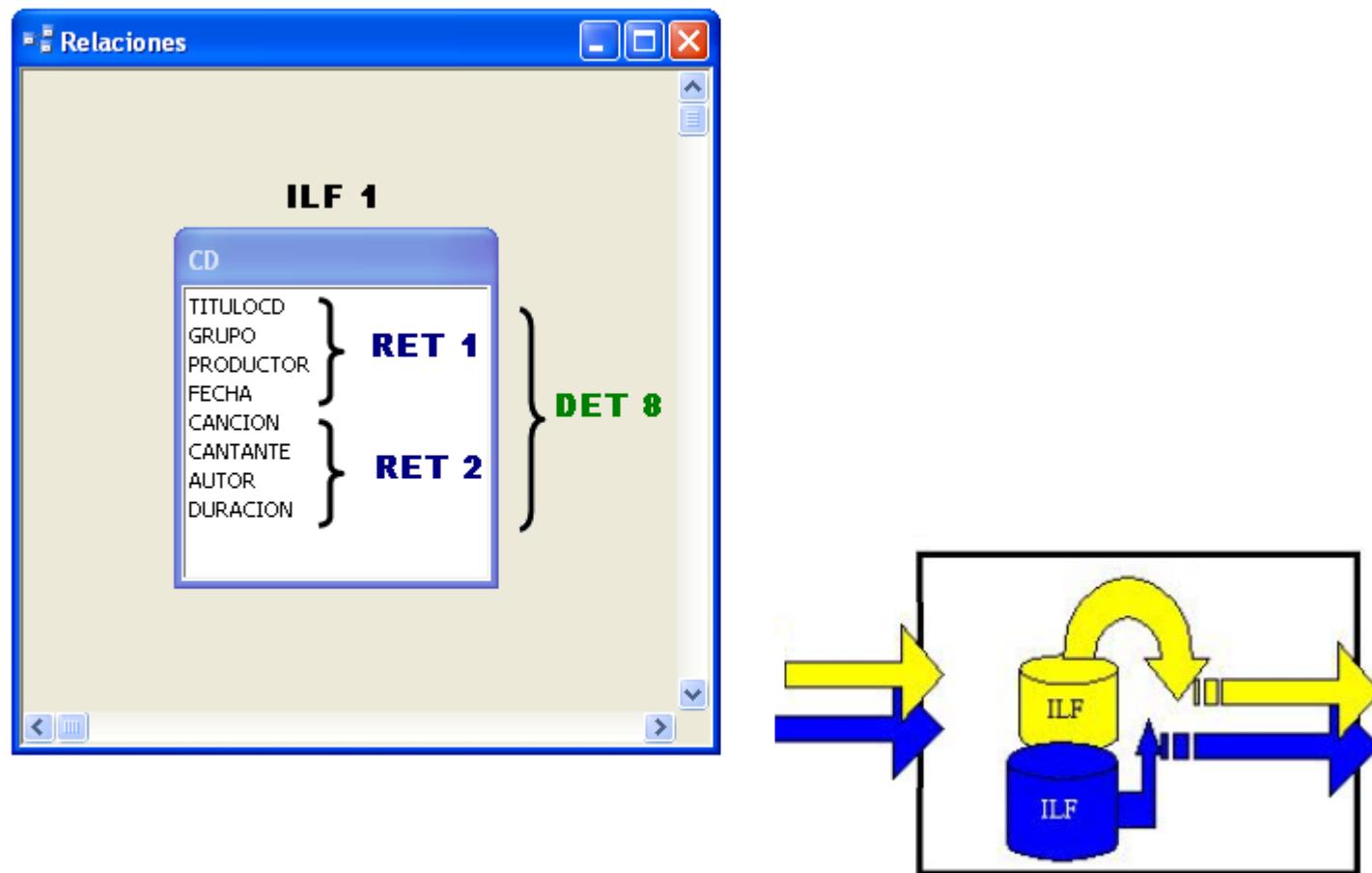
Nombre Proyecto	Puntos de Función	Lenguaje	Esfuerzo en horas
Sénia	200	COBOL	5.017
Paláncia	150	PASCAL	2.569
Turia	375	4GL	3.011
Albufera	500	PASCAL	9.479
Magro	425	4GL	3.342
Cabriel	800	PASCAL	13.349
Júcar	180	PASCAL	2.800
Serpis	325	4GL	2.541
Montnegre	225	PASCAL	4.528
Segura	470	COBOL	13.218

OTRAS UTILIDADES DE LOS PUNTOS DE FUNCIÓN.

- Comparar lo que solicitó el cliente con lo que recibió.
- Comparar la productividad de los diferentes entornos de desarrollo.
- Comparar la calidad que se obtiene mediante las diferentes técnicas de desarrollo.

CUENTA DE PUNTOS DE FUNCIÓN EN EL CASO PRÁCTICO

CONTEO DE PUNTOS DE FUNCION ARCHIVOS LÓGICOS INTERNOS (ILF)

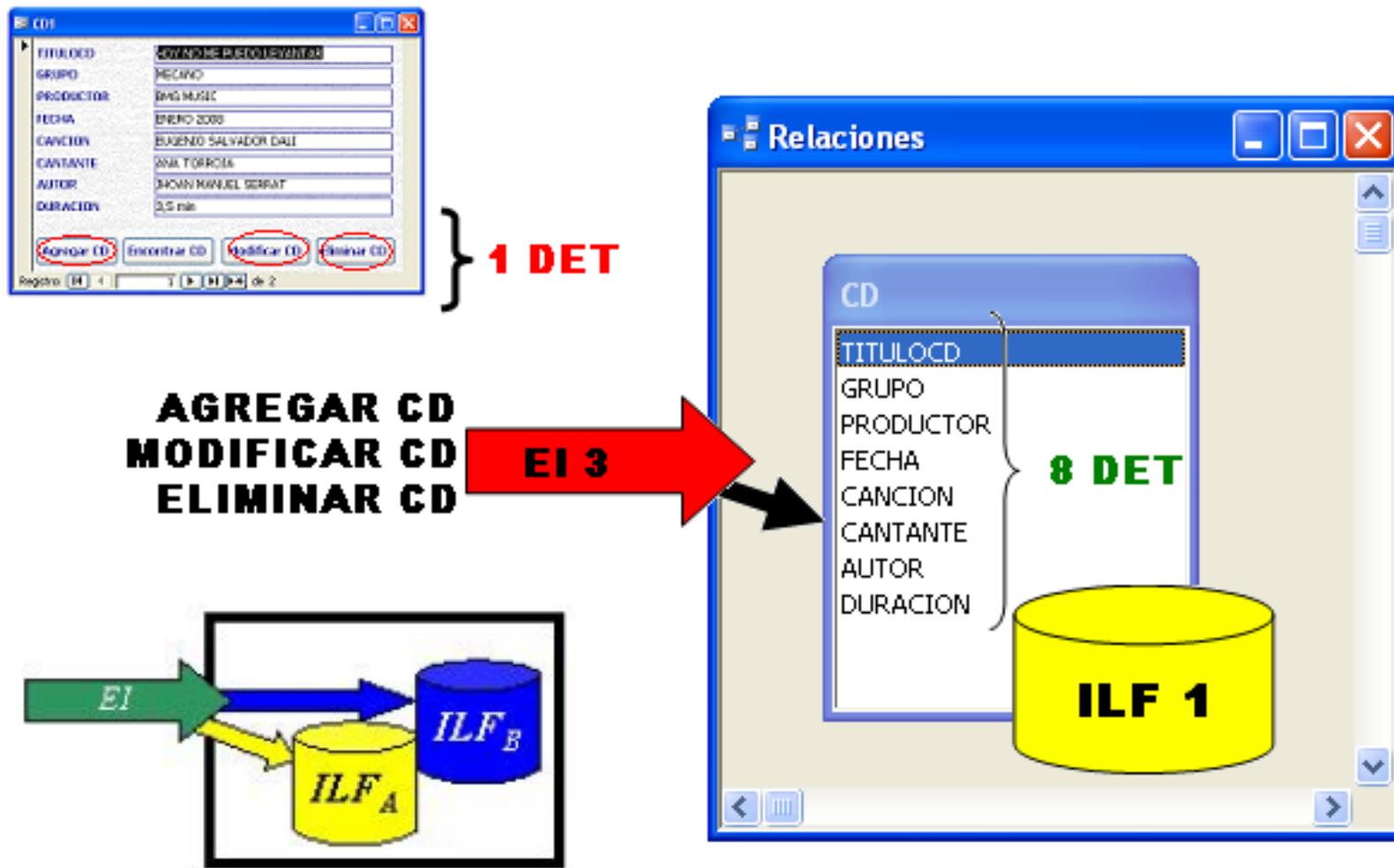


TOTAL ILF

Tipos de Registros	Elementos de Datos		
	1-19	20-50	>50
1	Baja	Baja	Media
2-5	Baja	Media	Alta
>5	Media	Alta	Alta

ILF	#DETs	#RETs	Complejidad
CD	8	2	Baja

CONTEO DE PUNTOS DE FUNCION ENTRADAS EXTERNAS (EI)

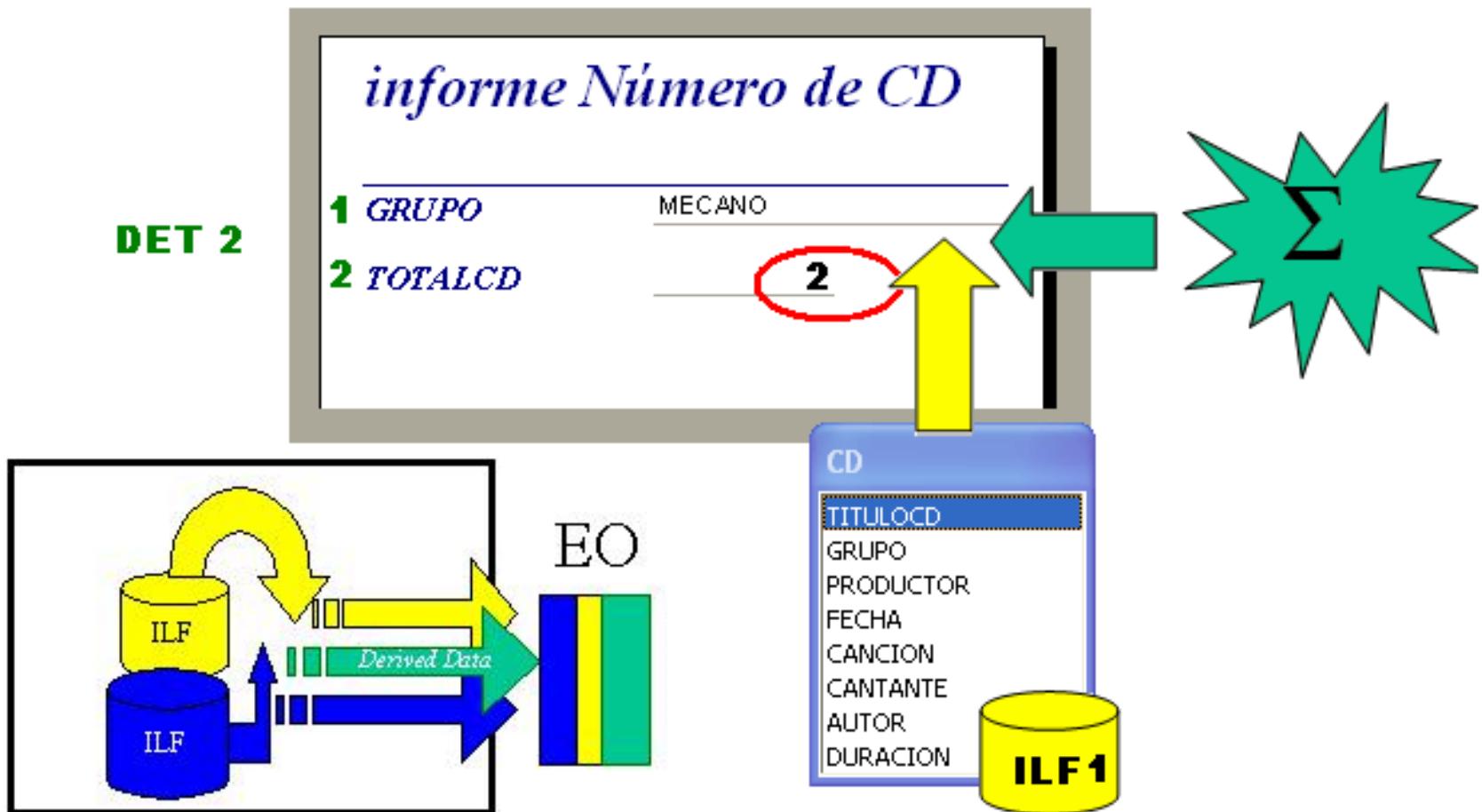


TOTAL EI

Archivos referenciados	Elementos de Datos		
	1-4	5-15	>15
0-1	Baja	Baja	Media
2	Baja	Media	Alta
3 ó más	Media	Alta	Alta

EI	#DETs	#FTRs	Complejidad
AGREGAR CD	10	1	Baja
MODIFICAR CD	9	1	Baja
ELIMINAR CD	3	1	Baja

CONTEO DE PUNTOS DE FUNCION SALIDAS EXTERNAS (EO)

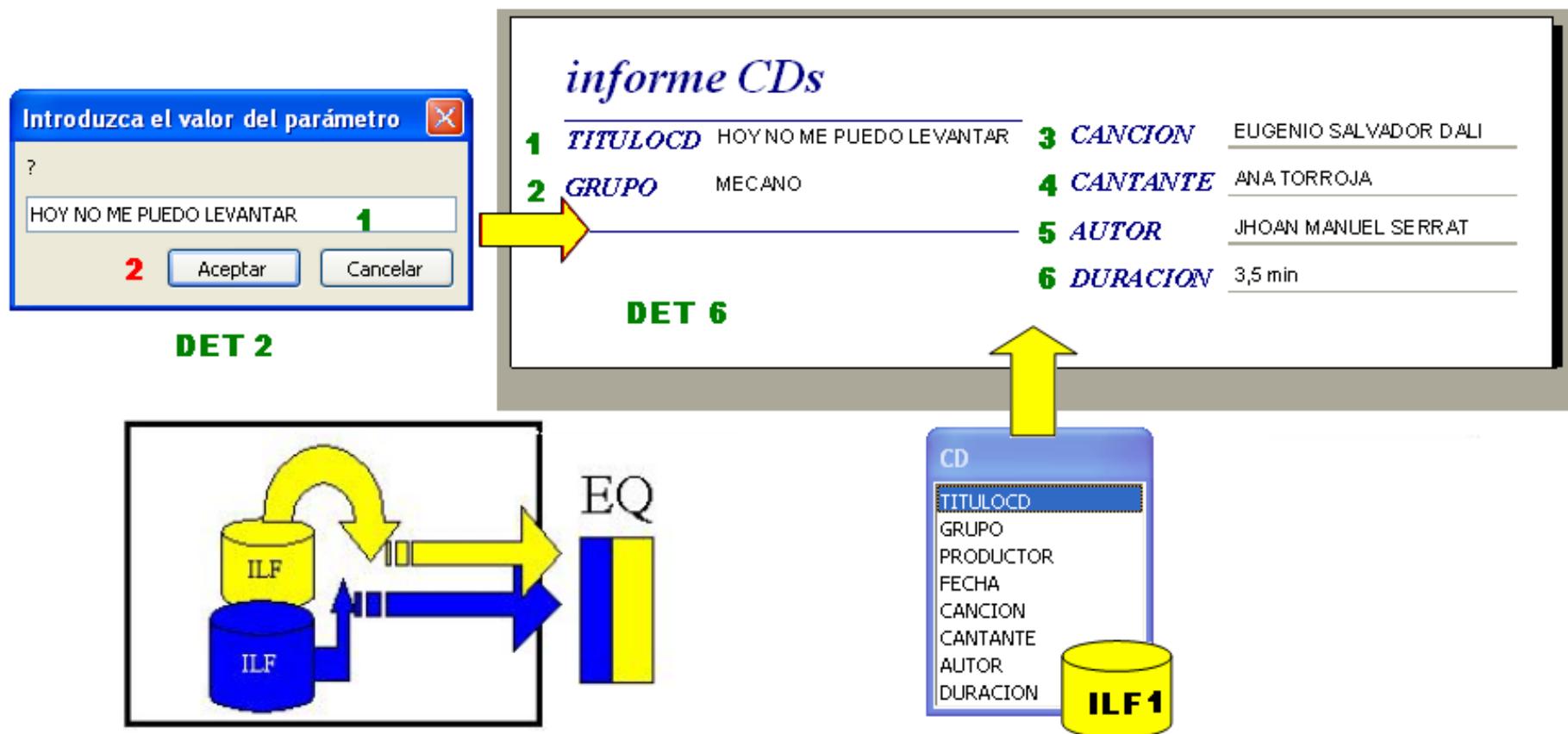


TOTAL EO

Archivos referenciados	Elementos de Datos		
	1-5	6-19	>19
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
>3	Media	Alta	Alta

EO	#DETs	#FTRs	Complejidad
OBTENER TOTAL CD	2	1	Baja

CONTEO DE PUNTOS DE FUNCION CONSULTAS EXTERNAS (EQ)



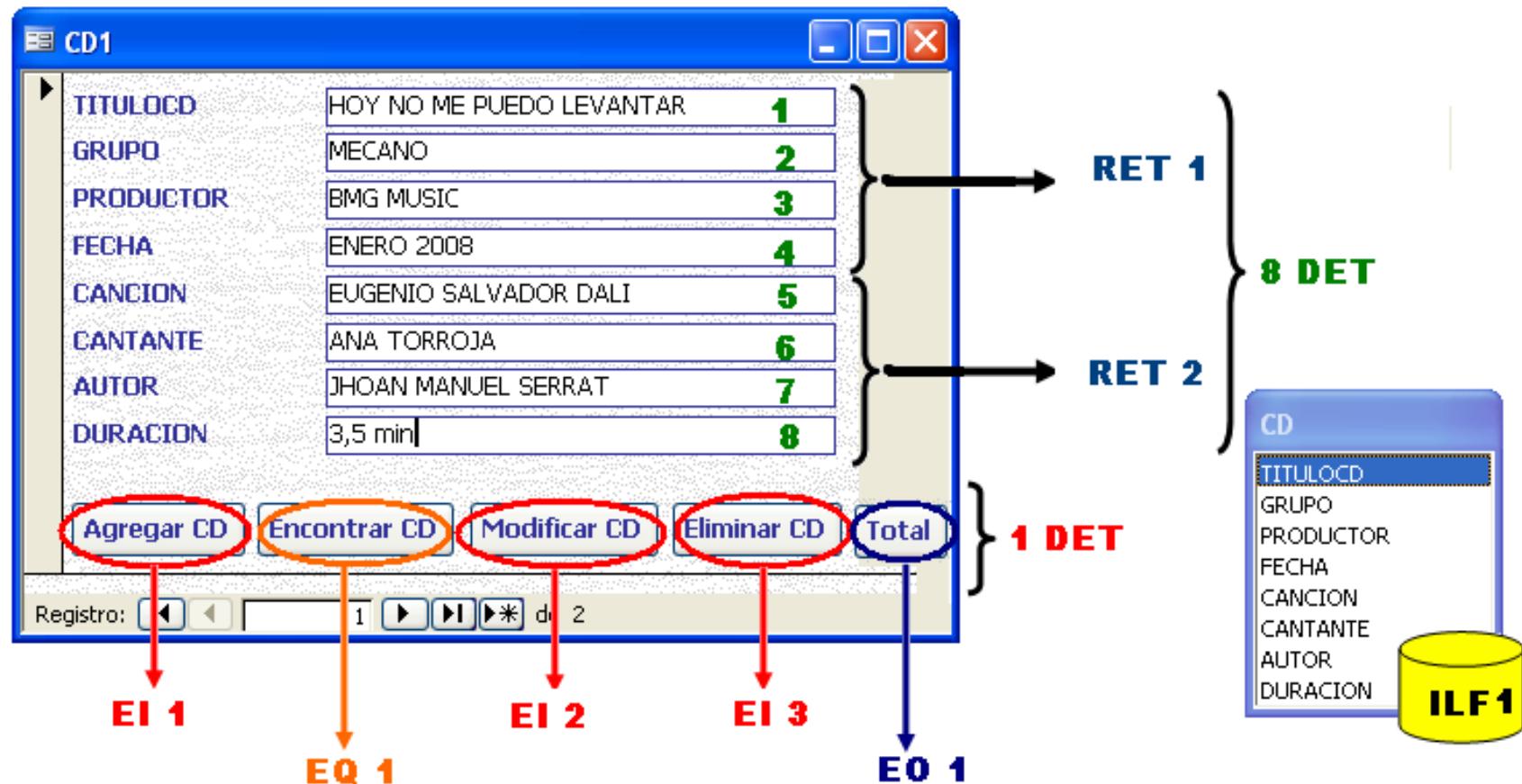
TOTAL EQ

Archivos referenciados	Elementos de Datos		
	1-5	6-19	>19
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
>3	Media	Alta	Alta

EQ	#DETs	#FTRs	Complejidad
(Entrada) ENCONTRAR CD	2	1	Baja
(Salida) ENCONTRAR CD	6	1	Baja

BAJA

RESUMEN PF



TOTAL PUNTOS DE FUNCION SIN AJUSTAR

	COMPLEJIDAD				APORTE
	BAJA	MEDIA	ALTA		
Entradas Externas (EI) →	3	0	0		9
Salidas Externas (EO) ←	1	0	0		4
Consultas Externas (EQ) ↔	1	0	0		3
Archivos Lógicos Internos (ILF)	1	0	0		7
Archivos de Interface Externa (EIF)	0	0	0		0
				TOTAL UPF	23

UNA VEZ OBTENIDOS LOS
PUNTOS DE FUNCIÓN, VAMOS A
DAR UN VISTAZO AL MODELO
COCOMO II

EL MODELO DE COSTO DISEÑO ANTICIPADO Y POST-ARQUITECTURA.

- Los modelos de Diseño Anticipado y Post-Arquitectura se basan en la misma filosofía a la hora de proporcionar una estimación.
- La fórmula básica para obtener una estimación de esfuerzo con ambos modelos es:

$$\mathbf{MP \text{ NOMINAL} = A \times (Size)B}$$

MP: Esfuerzo (meses-persona)

A: Constante

Size: Tamaño

B: Factor de escala

CONSTANTE A.

- La constante A, se usa para capturar los efectos multiplicativos de esfuerzo en proyectos de tamaño incremental.
- Provisionalmente se le ha estimado un valor de **2.94**

VARIABLE SIZE.

- Se calcula como:

$$\underline{\text{Size}} = \text{Size} \times \left[1 + \frac{\text{BRAK}}{100} \right]$$

- BRAK: Porcentaje de Rotura para ajustar el tamaño eficaz del producto. La rotura refleja la volatilidad de los requisitos en un proyecto. Es el porcentaje de código desperdiciado debido a la volatilidad de los requisitos.

EJEMPLO

- *Un proyecto que parte de 100.000 instrucciones (Size= 100.000) pero desecha el equivalente a 20.000 instrucciones tiene un valor de BRAK de 20.*
- *Esto debe usarse para ajustar el Tamaño efectivo del proyecto a 120.000 instrucciones para una estimación COCOMO II. (Size = 100.000 x [1 + (20/100)])*

Tamaño de la Aplicación (Size)

- Se mide en unidades de líneas de código fuente (KSLOC). ($K= 1000$ SLOC)
- Generalmente se opta por utilizar los **Puntos de Función sin Ajustar** para determinar el tamaño del proyecto, éstos deben convertirse en líneas de código fuente en el lenguaje de implementación (ensamblador, lenguajes de alto nivel, lenguajes de cuarta generación, etc...)

Ejemplo: Tabla de Conversión

Lenguaje	SLOC / UFP
Ada	71
AI Shell	49
APL	32
Assembly	320
Assembly (Macro)	213
ANSI/Quick/Turbo Basic	64
Basic – Compiled	91
Basic Interpreted	128
C	128
C++	29
Visual Basic	32
ANSI Cobol 85	91

$C++: 165 \text{ UNFP} \times 29 = 4785 \text{ SLOC}$

(Líneas de Código Fuente)

Haciendo la conversión: $4785/1000 = 4.785$

KSLOC (Miles de Líneas de Código Fuente).

Variable B (ahorro y gasto software de escala).

- Determina los gastos y ahorros relativos de escala encontrados en proyectos software de distinto tamaño.

5

$$B = 0.91 + 0.01 \times \sum_{j=1}^5 SF_j$$

Análisis de B

- **Si $B < 1.0$.** El proyecto presenta ahorros de escala. Si el tamaño del producto se dobla, el esfuerzo del proyecto es menor que el doble. La productividad del proyecto aumenta a medida que aumenta el tamaño del producto.
- **Si $B = 1.0$.** Los ahorros y gastos de escala están equilibrados.
- **Si $B > 1.0$.** El proyecto presenta gastos de escala. Se debe normalmente a dos factores principales: El crecimiento del gasto en comunicaciones y el gasto en crecimiento de la integración de un gran sistema. ($>Size + Personas + Gasto\ en\ comunicaciones$. Integrar un producto pequeño requiere esfuerzo de desarrollo del Producto pequeño + Gasto por esfuerzo de integración, mantenimiento y prueba con el otro Producto grande)

Factores de Escala COCOMOII de Diseño Anticipado

Factores de Escala (SF _j)	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PREC	Completa-mente sin precedentes	Prácticamen-te sin precedentes	Casi sin precedentes	Algo familiar	Muy familiar	Completa-mente familiar
FLEX	Riguroso	Relajación ocasional	Algo de relajación	Conformidad general	Algo de conformidad	Metas generales
RESL*	Poco (20%)	Algo (40%)	A menudo (60%)	Generalmen-te (75%)	En su mayor parte (90%)	Por completo (100%)
TEAM	Interacciones muy difíciles	Algo de dificultad en las interacciones	Interacciones básicamente cooperativas	Bastante cooperativo	Altamente cooperativo	Completas interacciones
PMAT	Peso medio de respuestas "Sí" para el cuestionario de Madurez CMM					

Valores de los Factores de escala

Factores de Escala (Wi)	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL*	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

Por ejemplo, si tenemos factores de escala con valores Muy Bajo, entonces un proyecto 100 KSLOC tendrá $SF_j = 31,62$, $B=1.2262$ y un esfuerzo relativo $E=100^{1.2262}=283 PM.$

(PREC) (FLEX). Precedencia y Flexibilidad de desarrollo.

Características	Muy Bajo	Nominal/Alto	Extra Alto
Precedencia			
Comprensión organizacional	General	Considerable	Profundo
Experiencia en trabajo con sistemas Sw relacionados	Moderado	Considerable	Extenso
Desarrollo concurrente de nuevo Hw asociado y procedimientos operacionales	Extenso	Moderado	Algo
Necesidad de arquitecturas de proceso de datos innovativas, algoritmos	Considerable	Algo	Mínimo
Flexibilidad de Desarrollo			
Necesidad de conformidad del Sw con requisitos preestablecidos	Completo	Considerable	Básico
Necesidad de conformidad del Sw con especificaciones de interfaz externas	Completo	Considerable	Básico
Prioridad en finalización anticipada	Alto	Medio	Bajo

(RESL) Arquitectura/Resolución de Riesgos

Características	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
El plan de gestión de riesgos identifica todos los items de riesgos críticos, establece hitos para resolverlos mediante PDR	Ninguno	Poco	Algo	Generalmente	A menudo	Completamente
Horario, presupuesto e hitos internos con PDR compatible con el Plan de gestión de riesgos	Ninguno	Poco	Algo	Generalmente	A menudo	Completamente
Tanto por ciento de horario desarrollado dedicado a establecer la arquitectura dados los objetivos generales del producto	5	10	17	25	33	40
Porcentaje de arquitectos Sw de alto nivel requeridos, disponibles para el proyecto	20	40	60	80	100	120
Herramientas de soporte disponibles para resolver items de riesgo, desarrollar y verificar garantías de la arquitectura	Ninguno	Poco	Algo	Bueno	Fuerte	Completo
Nivel de incertidumbre en drivers de arquitectura claves: misión, interfaz de usuario, COTS, Hw, tecnología, ejecución	Extremo	Significativo	Considerable	Algo	Poco	Muy Poco
Número y criticalidad de items de riesgo	>10 Crítico	5-10 Crítico	2-4 Crítico	1 Crítico	>5 No Crítico	<5 No Crítico

(TEAM). Cohesión del Equipo

Características	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Consistencia de objetivos y culturas	Poco	Algo	Básico	Considerable	Fuerte	Completo
Habilidad y servicialidad para acomodar objetivos de otros grupos	Poco	Algo	Básico	Considerable	Fuerte	Completo
Experiencia de los Desarrollados en operar como un equipo	Nada	Poco	Poco	Básico	Considerable	Extensa
Para lograr visión compartida y compromisos	Nada	Poco	Poco	Básico	Considerable	Extensa

Tabla 7.11. TEAM Componentes de medida

(PMAT). Madurez del proceso

Nivel de Madurez Global

- Nivel 1 CMM (Mitad inferior)
- Nivel 1 CMM (Mitad superior)
- Nivel 2 CMM
- Nivel 3 CMM
- Nivel 4 CMM
- Nivel 5 CMM

Áreas de Proceso	Frecuencia		En la Mitad (40-60%)	Ocasional- mente (10-40%)	En Pocas Ocasiones (<10%)	No se Aplica	No se Conoce
	Clave	Casi siempre (>90%)	mente (60-90%)				
1 Gestión de requisitos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 Planificación de proyectos Software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 Seguimiento de proyecto Software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Gestión de subcontrato Software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5 Aseguramiento de la calidad Software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6 Gestión de la configuración Software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	

$$5 - \left[\sum_{i=1}^{18} \left[\frac{\text{KPA}\%_i}{100} \times \frac{5}{18} \right] \right]$$

Ejemplo Tabla Valoración PMAT

AJUSTE MEDIANTE DRIVERS DE COSTE

- Los drivers de coste se usan para capturar características del desarrollo del software que afectan al esfuerzo para completar el proyecto.
- Estos valores pueden ir desde Extra Bajo hasta Extra Alto.
- Cada nivel de medida de cada driver de coste tiene un peso asociado. El peso se llama multiplicador de esfuerzo (**EM**).
- Si $EM = 1$ Esfuerzo Nominal
- Si $EM > 1$ Mayor Esfuerzo de Desarrollo
- Si $EM < 1$ Menor Esfuerzo de Desarrollo

AJUSTE MEDIANTE DRIVERS DE COSTE

- Los **EM** se usan para ajustar el esfuerzo Meses-persona nominal.
- Hay 7 Multiplicadores de esfuerzo para el Modelo de Diseño Anticipado y 17 para el modelo de Post-Arquitectura .

$$\textbf{MP} = A \times (\text{Size})^B \times \prod \underline{\textbf{EM}_i}$$

Multiplicadores de esfuerzo del Diseño Anticipado y Post-Arquitectura

Drivers de coste del Diseño Anticipado	Drivers de coste combinados, homólogos del Post-Arquitectura
RCPX	RELY, DATA, CPLX, DOCU
RUSE	RUSE
PDIF	TIME, STOR, PVOL
PERS	ACAP, PCAP, PCON
PREX	AEXP, PEXP, LTEX
FCIL	TOOL, SITE
SCED	SCED

- (RCPX). Fiabilidad del Producto y Complejidad
- (RUSE) Reutilización Requerida
- (PDIF) Dificultad de la Plataforma
- (PREX) Experiencia Personal
- (FCIL) Facilidades
- (SCED) Planificación Temporal

Multiplicadores de Esfuerzo – Diseño anticipado

	Extra Bajo	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
RCPX	0.73	0.81	0.98	1.00	1.30	1.74	2.38
RUSE	--	--	0.95	1.00	1.07	1.15	1.24
PDIF	--	--	0.87	1.00	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.00	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.00	0.87	0.73	0.62
SCED	--	1.43	1.14	1.00	1.00	1.00	--

	Extra Bajo	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Énfasis en la fiabilidad, documentación	Muy Poco	Poco	Algo	Básico	Fuerte	Muy Fuerte	Extremo
Complejidad del producto	Muy Simple	Simple	Algo	Moderado	Complejo	Muy Complejo	Extremadamente Complejo
Medida de la Base de Datos	Pequeño	Pequeño	Pequeño	Moderado	Grande	Muy Grande	Muy Grande

niveles de medida de RCPX

	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
RUSE		Nada	A lo largo del programa	A lo largo del proyecto	A lo largo de la línea de producto	A lo largo de múltiples líneas de producto

Tabla 7.15. Resumen del nivel de medida RUSE

	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Restricciones de tiempo y de almacenamiento	≤ 50%	>50%	65%	80%	90%
Volatilidad de la Plataforma	Muy Estable	Estable	Volátil	Volátil	Volátil

Tabla 7.16. Niveles de medida PDIF

	Extra Bajo	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Experiencia en aplicaciones, plataforma, lenguaje y herramienta	≤ 3 Meses	5 Meses	9 Meses	1 Año	2 Años	4 Años	6 Años

Tabla 7.17. Niveles de medida PREX

	Extra Bajo	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Soporte de TOOL	Mínimo	Algo	Herramienta CASE simple	Herramientas de ciclo de vida básicas	Bueno; moderado	Fuerte; moderado	Fuerte; Bien integrado
Condiciones Multilugar	Soporte débil de desarrollo multilugar complejo	Algo de soporte de desarrollo multilugar complejo	Algo de soporte de desarrollo multilugar moderadamente complejo	Soporte básico de desarrollo multilugar moderadamente complejo	Fuerte soporte de desarrollo multilugar moderadamente complejo	Fuerte soporte de desarrollo multilingue simple	Soporte muy fuerte de desarrollo multilingue simple

Tabla 7.18. Niveles de medida FCIL

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

Tabla 7.19. Resumen de nivel de medida SCED

Ejemplo

- Si al analizar los drivers de coste para nuestro proyecto obtenemos los siguientes resultados:

	MB	B	N	A	MA	EA
RCPX				x		
RUSE			x			
PDIF			x			
PERS			x			
PREX				x		
FCIL				x		
SCED			x			

EB=Extra Bajo
MB=Muy Bajo
B=Bajo
N=Nominal
A=Alto
MA=Muy Alto
EA=Extra Alto

$$\text{II } EM_i = 1.30 \times 1 \times 1 \times 1 \times 0.87 \times 0.87 \times 1.0 = 0.984$$

OBTENCIÓN DE DRIVERS DE COSTE PARA MODELO DE DISEÑO POSTARQUITECTURA

- Idem....



VALORES DE TIEMPO DE DESARROLLO.

- La ecuación siguiente es la ecuación inicial de tiempos base para las tres etapas de COCOMO II:

$$TDEV = [3.67 \times PM^{(0.28+0.2 \times (B-1.01))}] \times \frac{SCED\%}{100}$$

TDEV: es el tiempo en meses desde la determinación de una línea base de requisitos del producto hasta que se completa una actividad de aceptación que certifica que el producto satisface los requisitos.

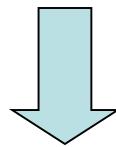
B: es la suma de los factores de escala del proyecto.

SCED: % es el porcentaje de compresión/expansión en el multiplicador de esfuerzo SCED,(Tabla anterior)

**APLICAMOS EL MODELO
COCOMOII AL CASO DE
ESTUDIO**

TRANSFORMACIÓN PF A SLOC EN EL CASO PRÁCTICO

Lenguaje	SLOC / UPF
Ada	71
Al Shell	49
APL	32
Assembly	320
Assembly (Macro)	213
ANSI/Quick/Turbo Basic	64
Basic - Compiled	91
Basic Interpreted	128
C	128
C++	29
Visual Basic	32


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

Size = 32 (VB) x 23 UPF = 736 SLOC = 0.736 KSLOC

CALCULO DE ESFUERZO NOMINAL

- Ecuación COCOMO con resultados:

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

- $PM_{\text{nominal}} = 2.94 \times (0.736)^?$

AJUSTE DE FACTORES DE ESCALA (B)

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

CALCULO DE ESFUERZO NOMINAL

- Ecuación COCOMO con resultados:

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

- $PM_{nominal} = 2.94 \times (0.736)^{1.05}$
- $PM_{nominal} = 2.13$ Meses-persona

AJUSTE DE DRIVERS DE COSTO

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

$\Pi(ME_i)$

CALCULO DE ESFUERZO AJUSTADO

- Ecuación COCOMO con resultados:

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

- $PM_{ajustado} = 2.13 \times 1.004$
- $PM_{ajustado} = 2.14$ Meses-persona

ANÁLISIS DEL ESFUERZO

- 2 PM = 2 Personas trabajando 1 Mes?
- 2 PM = 1 Persona trabajando 2 Meses?
- 9 PM = 9 Personas trabajando 1 Mes?
- 9 PM = 3 Personas trabajando 3 Meses?
- 9 PM = 1 Persona trabajando 9 Meses?
- Si el proyecto es tener un hijo?

CÁLCULO DEL TIEMPO DE DESARROLLO

$$TDEV = [3.67 \times PM^{(0.28+0.2x(B-1.01))}] \times \frac{SCED\%}{100}$$

$$TDEV = [3.67 \times 2.14^{(0.278+0.2x(1.05-1.01))}] \times 1$$

$$\begin{aligned} TDEV &= [3.67 \times 2.14^{0.286}] \times 1 = 4.56 \text{ Meses} \\ &= 4.6 \text{ Meses} \end{aligned}$$

CÁLCULO MEDIANTE HERRAMIENTA COCOMO

E:\coradmo\coradmojem.est - USC-COCOMO II.1999.0

File Edit View Parameters Calibrate Phase Maintenance Help

Project Name: Sistema CD

Development Model: Early Design

X	Module Name	Module Size	LABOR Rate (\$/month)	EAF	NOM Effort DEV	EST Effort DEV	PROD	COST	INST COST	Staff	RISK
	ITERACION 1	F:736	700.00	1.00	2.1	2.1	344.6	1494.97	2.0	0.5	0.0

Total Lines of Code:	Estimated	Effort	Sched	PROD	COST	INST	Staff	RISK
736	Optimistic	1.4	4.1	514.4	1001.63	1.4	0.3	
	Most Likely	2.1	4.6	344.6	1494.97	2.0	0.5	0.0
	Pessimistic	3.2	5.3	229.7	2242.46	3.0	0.6	

Main.csv & Phase.csv are saved in E:\coradmo\COCOMORAD\

ANALISIS DE RESULTADOS

- El Tiempo de desarrollo y costo obtenido se ajusta a la realidad?
(E = 2.1 PM , T = 4,6 Meses, C= 1494 USD
700xM)



ES NECESARIO DECIDIR

- El modelo de estimación utilizado **NO** se ajusta a la realidad, hay que buscar otras alternativas.
- Buscar alternativas muchas veces implica el hacer un análisis más exhaustivo del modelo empleado y hallar posibles limitaciones y alternativas de adaptación.



LIMITACIONES DEL MODELO COCOMOII

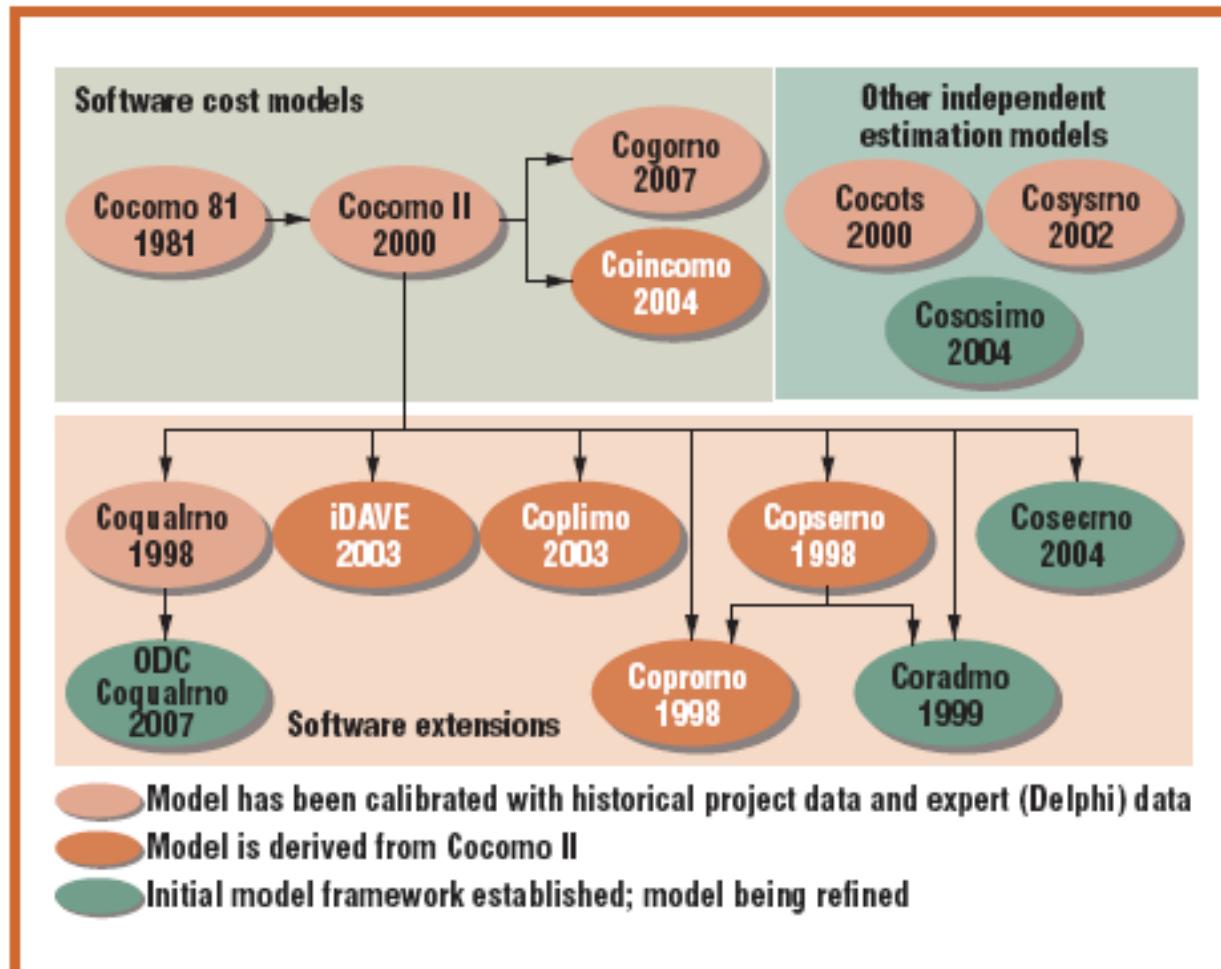
- Los factores cualitativos son difíciles de modelar.
- La calibración del modelo ha sido realizada con proyectos exitosos.
- Es fácil manipular el modelo para obtener los datos deseados.

LIMITACIONES DE LA APLICACIÓN DE COCOMO II AL CASO DE ESTUDIO

- Está basado en un ciclo de vida en cascada. (hemos elegido un modelo en EVOLUTIVO – espiral basado en la construcción de un prototipo)
- El modelo NO sirve para proyectos pequeños, esfuerzo < 16 PM, menos de 2 programadores por año (nuestro proyecto es pequeño).



ALTERNATIVAS DERIVADAS DE COCOMO



The Cocomo suite of models. Dates indicate the time that the first paper was published for the model.

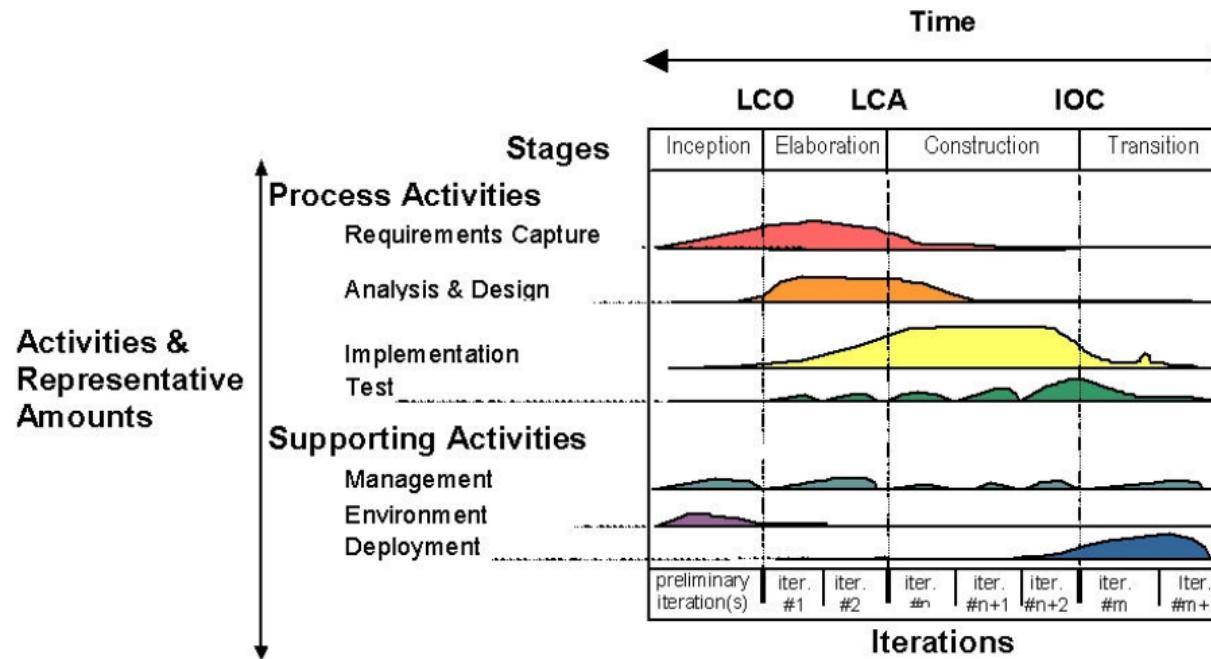
Modelo COPSEMO

(Constructive Phased Schedule & Effort Model)

- Se centra en el costo de desarrollo de software (menor de 64 meses-persona) mediante una mayor complejidad de cálculo para el bajo esfuerzo.
- Asume que los datos se encuentran en el Modelo COCOMO II. No tiene drivers diferentes.
- El modelo permite aplicarse a las distintas fases especificando el esfuerzo y el calendario en porcentajes.

Modelo COPSEMO

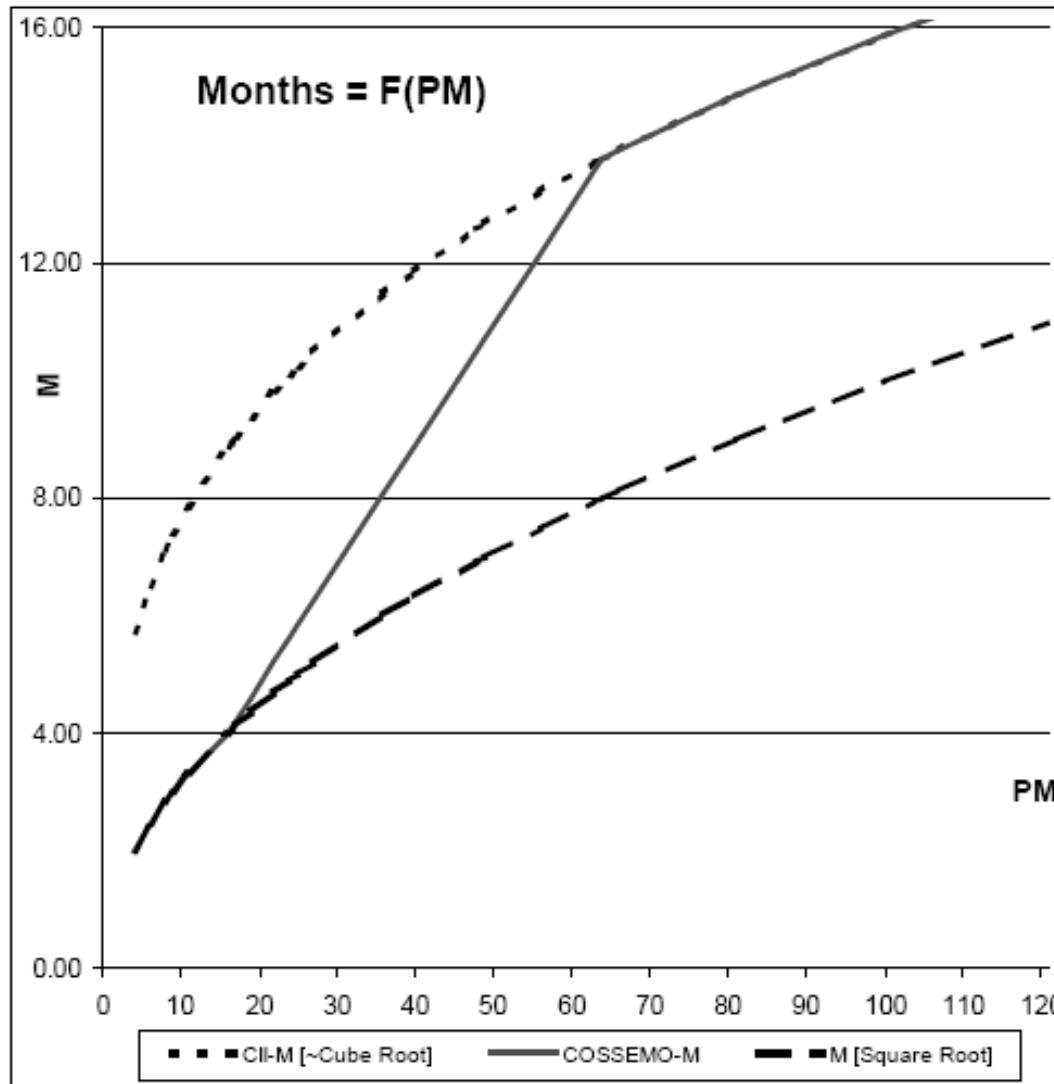
- Utiliza el MBSE (Model-Based Systems Engineering) y RUP (Proceso Unificado, se basa en un ciclo de vida evolutivo).
- Aunque está desarrollado como parte del modelo CORADMO, COPSEMO es reconocido como un modelo independiente que se utilizará como base para otros modelos de extensiones COCOMO II.



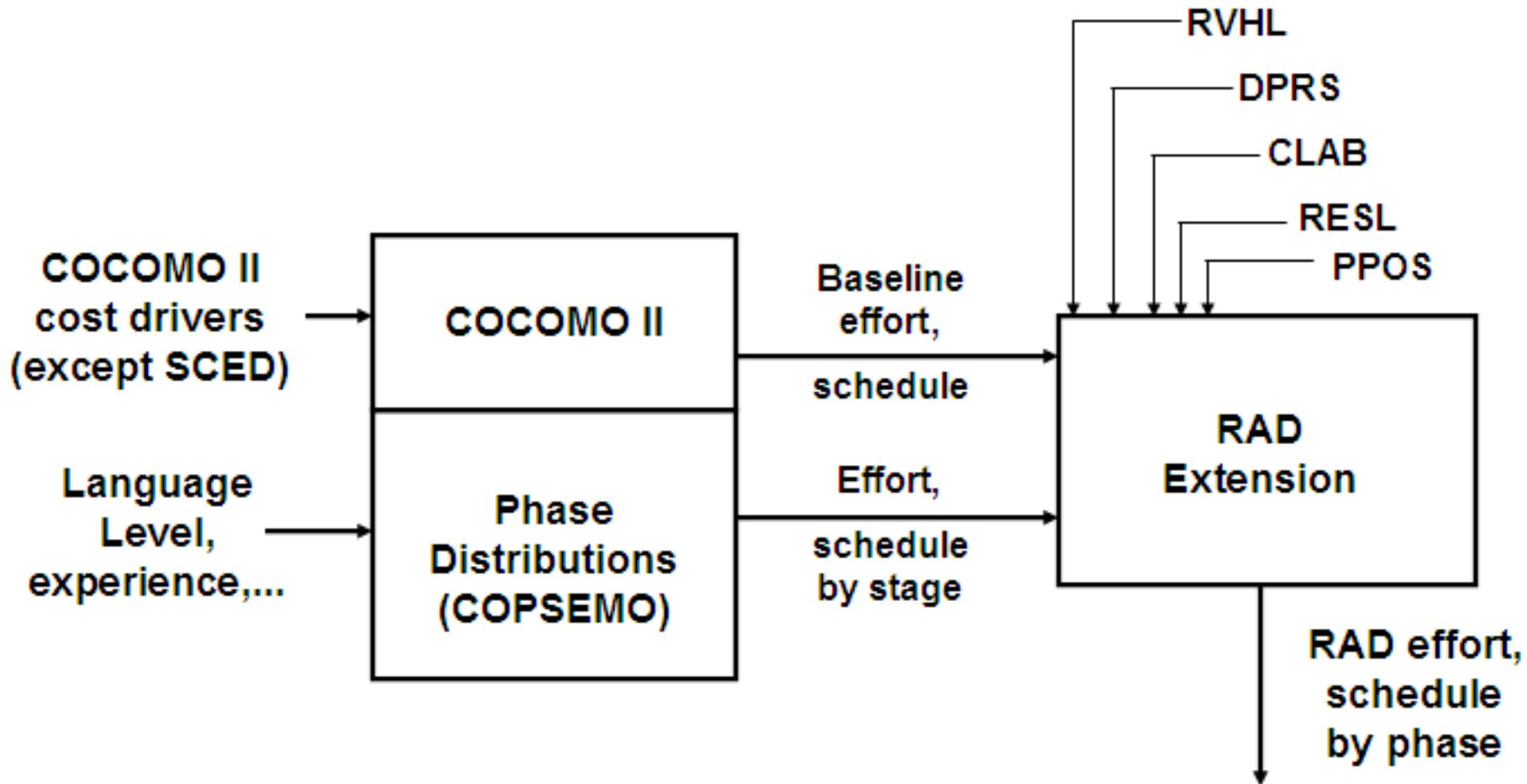
Modelo CORADMO (COCOMO RAD MODEL)

- Aplicable a proyectos basados en estrategias de desarrollo rápido de aplicaciones (RAD).
- Introduce 5 nuevos indicadores:
 - **RVHL** (Reutilización y Lenguajes de Muy Alto Nivel)
 - **DPRS** (Reingeniería en el proceso de desarrollo)
 - **CLAB** (Eficiencia en la colaboración)
 - **PPOS** (Preposicionamiento de activos)
 - **RESL** (Arquitectura y Resolución de Riesgos = COCOMOII).
- Orientado a proyectos pequeños, esfuerzo < 16PM
- Tiempo:
 - CORADMO= $\sqrt{2}$ (PM)
 - COCOMO Tiempo= $3^* \sqrt{3}$ (PM)

COMPARACIÓN DE CÁLCULO DE TIEMPO COCOMO vs COPSEMO-CORADOMO



Modelo Lógico CORADMO



APLICACION DE CORADMO AL CASO DE ESTUDIO

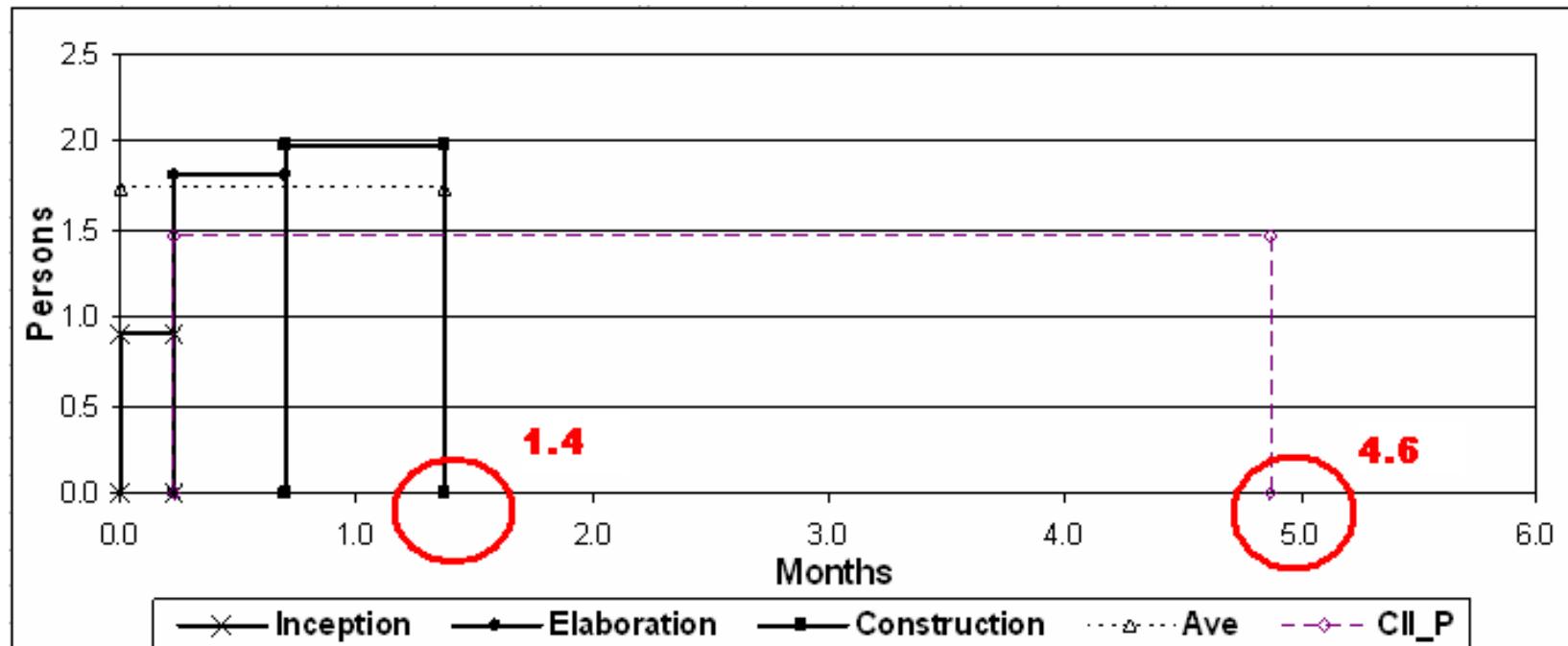
Construtive RAD Schedule estimation Model (CoRADSMo) Development

Sheet: ED&PO-Dev

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA			
1		CORADMO																												
2	Step																													
3	1.0	Currently implemented only for projects (not in modules)																											
4	Get COCOMO-II.98 data																													
5	2.0	BOLD: required	<i>italic: optional</i>		Project:	Sistema CD																								
6		(automatically from COCOMO.xls)																												
7	Including Scale Factor Ratings																													
13																														
14	3.0	Start with M_BS (i.e. After applying COSSEMO's M=f(PM) to COCOMO-II.1998 Effort (PM_C)) and P_BS calculated from PM_C/M_BS.																												
20		M_BS=	1.5		P_BS=	1.4614																								
21																														
22	4.0	Get Eff% & Sched % per stage																												
23	% Effort	10.0	<input type="button" value="▲"/>		% Schedul	20.0	<input type="button" value="▲"/>			% Effort	40.0	<input type="button" value="▲"/>			% Schedul	40.0	<input type="button" value="▲"/>			% Effort	60.0		% Schedule	60.0						
24	Inception		<input type="button" value="▼"/>		Inception		<input type="button" value="▼"/>			Elaboration		<input type="button" value="▼"/>			Elaboration		<input type="button" value="▼"/>			Construction			Construction							
25																														
26	5.0	Distribute PM (PM_C) and M (M_BS) according to the Effort% & Schedule% per stage; and calculate P for the stage from its PM/M.																												
36		Inception			Elaboration					Construction					Total E&C					Total										
37	Effort %	10.0				40.0				60.0						100.0				110.0										
38	Schedule %	20.0				40.0				60.0						100.0				120.0										
39	P/Ave(P)	0.50				1.00				1.00						1.00				Does not apply										
40	PM/M=P	PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P		PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P		PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P	PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P		PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P_ave	
41	BS	736	0.21	0.29	0.73		0.85	0.58		1.46	1.28	0.88	1.46							2.1	1.5	1.5	2.3	1.8	1.3					
42																														
43	6.0	Get the Schedule Multipliers values.																												
44																														
45																														
46		Inception			Elaboration				Construction																					
47		PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P		PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P		PM	<input type="button" value="▼"/>	M	<input type="button" value="▼"/>	P												
48	N	RVHL	0.980	0.980	1.000		0.99	0.99		1.00	1.00	1.00	1.00																	
49	N	DPRS	1.000	1.000	1.000		1.000	1.000		1.000	1.000	1.000	1.000																	
50	H	CLAB	0.930	0.930	1.000		0.950	0.950		1.000	0.980	0.980	1.000																	
51	H	RESL	1.000	1.000	1.000		1.000	1.000		1.000	1.000	1.000	1.000																	
52	VH	PPOS	1.060	0.860	1.233		1.060	0.860		1.233	1.060	0.860	1.233																	
53		II	0.966	0.784	1.233		0.997	0.809		1.233	1.039	0.767	1.354																	
54																														

CALCULO DEL ESFUERZO Y TIEMPO – CASO DE ESTUDIO

- Tiempo estimado: **1.4** Meses (4.6 Meses – COCOMO)
- Esfuerzo: 2.14 PM (PF: 23) (SLOC 736) (Calculado con COCOMO)

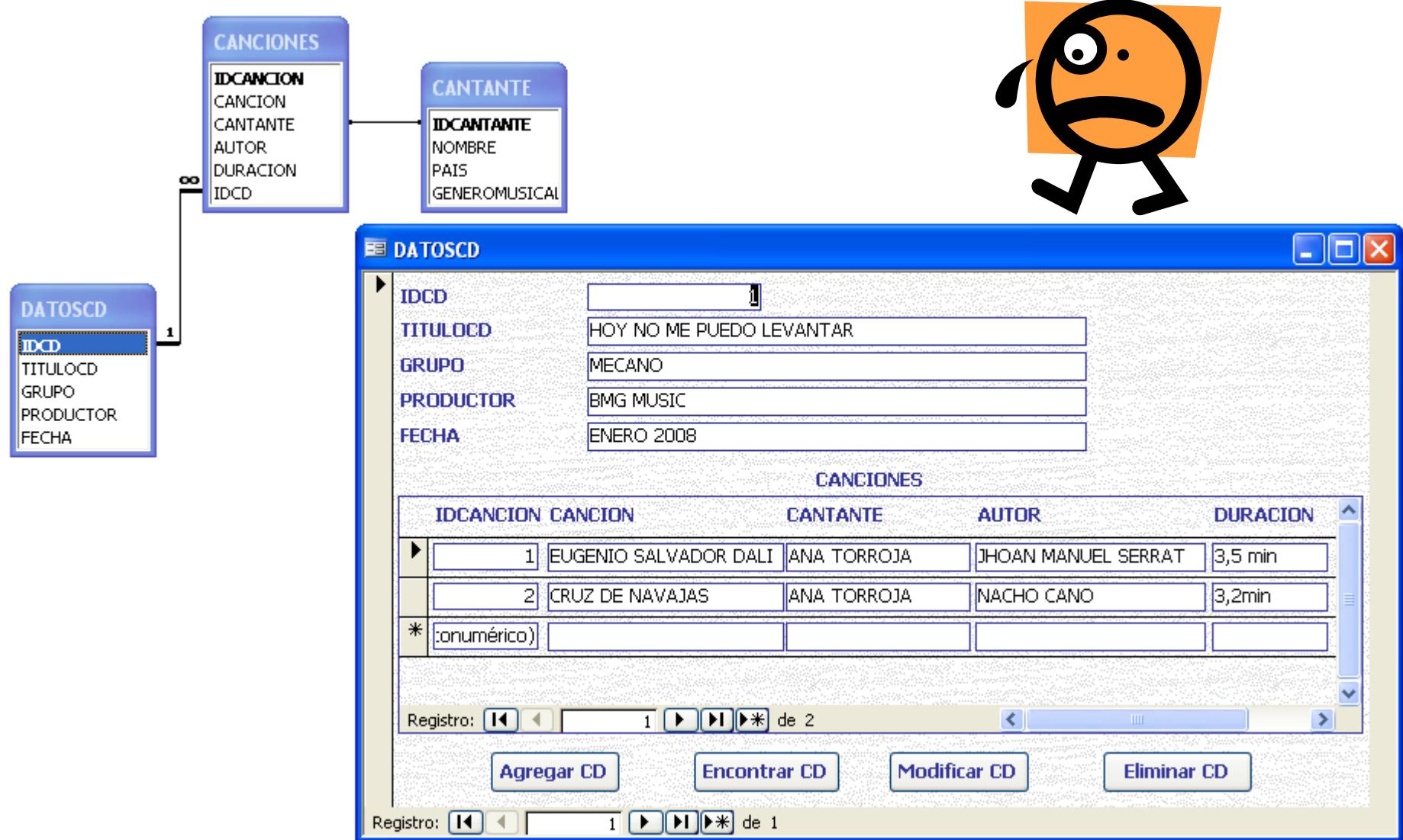


ANALISIS DE RESULTADOS

- De acuerdo a la experiencia, se podría concluir que el modelo CORADMO mejora la estimación, aunque esto debería igualmente validarse.
- Los PF introducen un grado de subjetividad en función de cómo se entienda los requisitos. (Aprox 30% de variación en la misma organización)
- La experiencia demuestra que para un mismo modelo Entidad Relación, se pueden obtener diferentes PF.
- La diferencia en resultados conduce a estimaciones de esfuerzo, costo y plazos drásticamente distintas.
- Para realizar el cálculo de los puntos de función debe existir un consenso dentro de la organización para el tratamiento de los requisitos, además de realizar un pre-diseño para que este tan sea una buena estimación del tamaño real de la aplicación.



SIGUIENTE ITERACIÓN



LA DURA REALIDAD

- **Creencias habituales:**



- No hay un método para desarrollar software que no sea ponerse delante de una pantalla y empezar a programar.
- Nuestro trabajo consiste en escribir un programa y hacer que funcione como sea.
- La documentación es esa idea de los jefes para perder tiempo e incumplir los plazos de manera justificada.
- No existe Ingeniería del Software, es una moda. El desarrollar Software es una arte. La reusabilidad y los estándares recortan la libertad de los informáticos.

PROBLEMAS DERIVADOS

- Mantenimiento de alto costo y alto riesgo

- Gran dependencia del individuo.

- Incumplimiento de plazos de entrega.

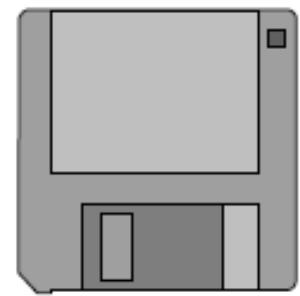
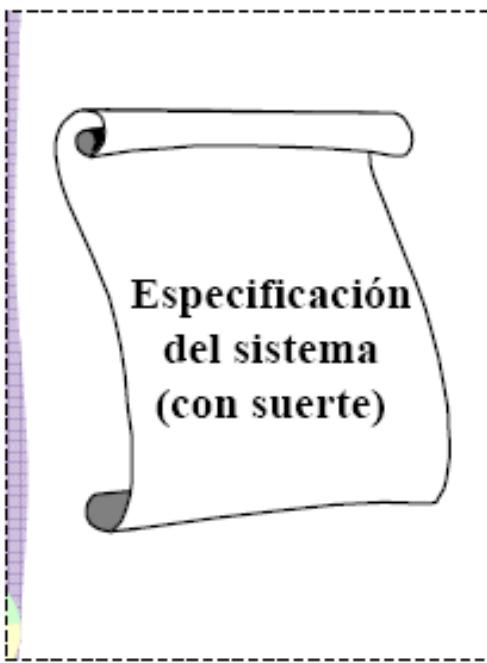
- No se tiene tiempo de recoger datos sobre el proceso de desarrollo de software que permitan estimaciones y planificaciones fiables.

- Insatisfacción de los usuarios con el producto terminado (cuando se termina).

- Dudosa calidad del software desarrollado.

- Poca importancia a las pruebas.





5. LA REALIDAD DEL DESARROLLO DE SOFTWARE EN EL ECUADOR

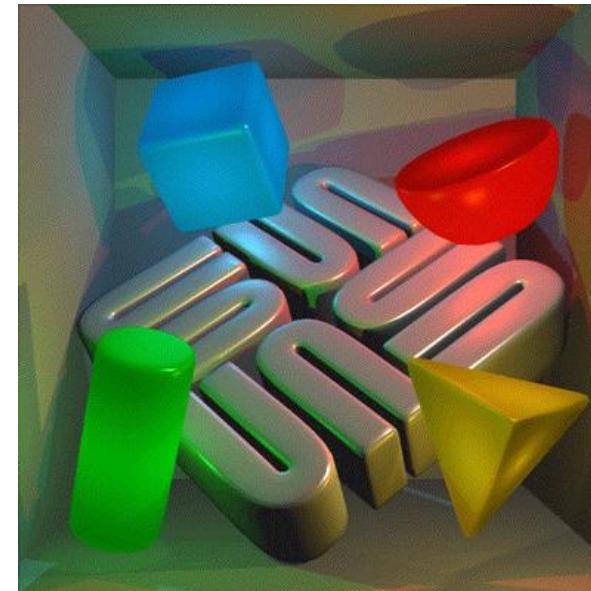
- Definición del Sector de mercado**

El sector de mercado al cual estamos orientados son empresas desarrolladoras de SW a nivel nacional, que realicen proyectos de Gestión o Especializados.

Como base de muestreo para el análisis de mercado se ha considerado varias empresas de las ciudades principales del país Guayaquil y Quito.

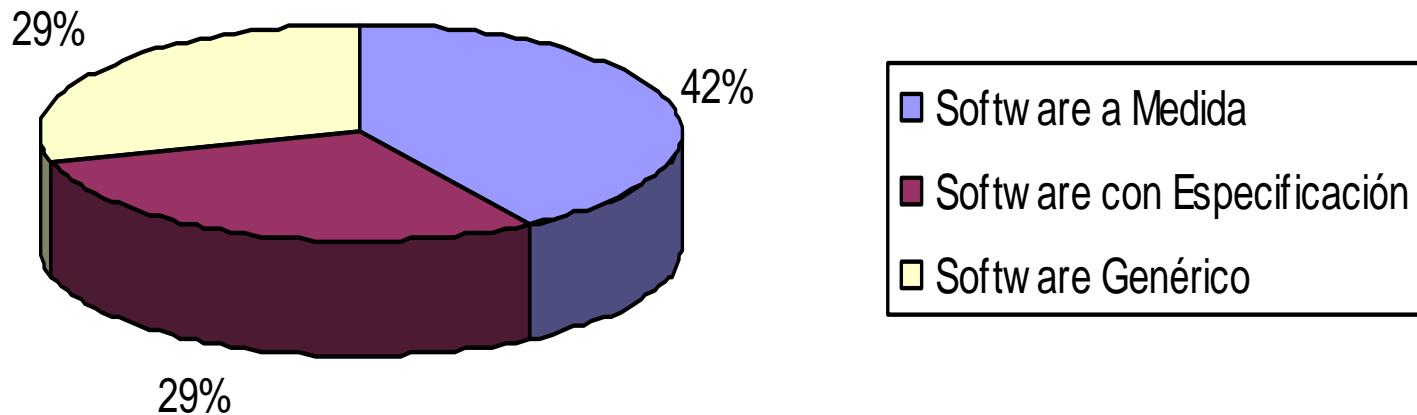
EMPRESAS DE SOFTWARE DEL ECUADOR ENTREVISTADAS

- **Intergrupo**
- **Creative Works**
- **Extremo Software**
- **Infoelect**
- **Kruger Corporation**
- **Logicstudio**
- **Macosa**
- **Sdconsult**
- **Vimeworks**
- **Represensa**
- **Corporación Latinoamericana De Software**
- **Agrosoft S.A.**



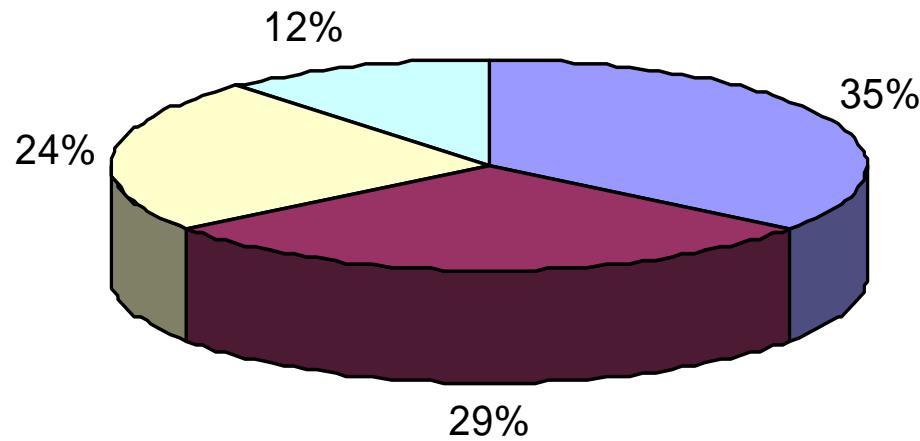
ESTADÍSTICAS

TIPOS DE PROYECTOS DE SOFTWARE DESARROLLADOS



ESTADÍSTICAS

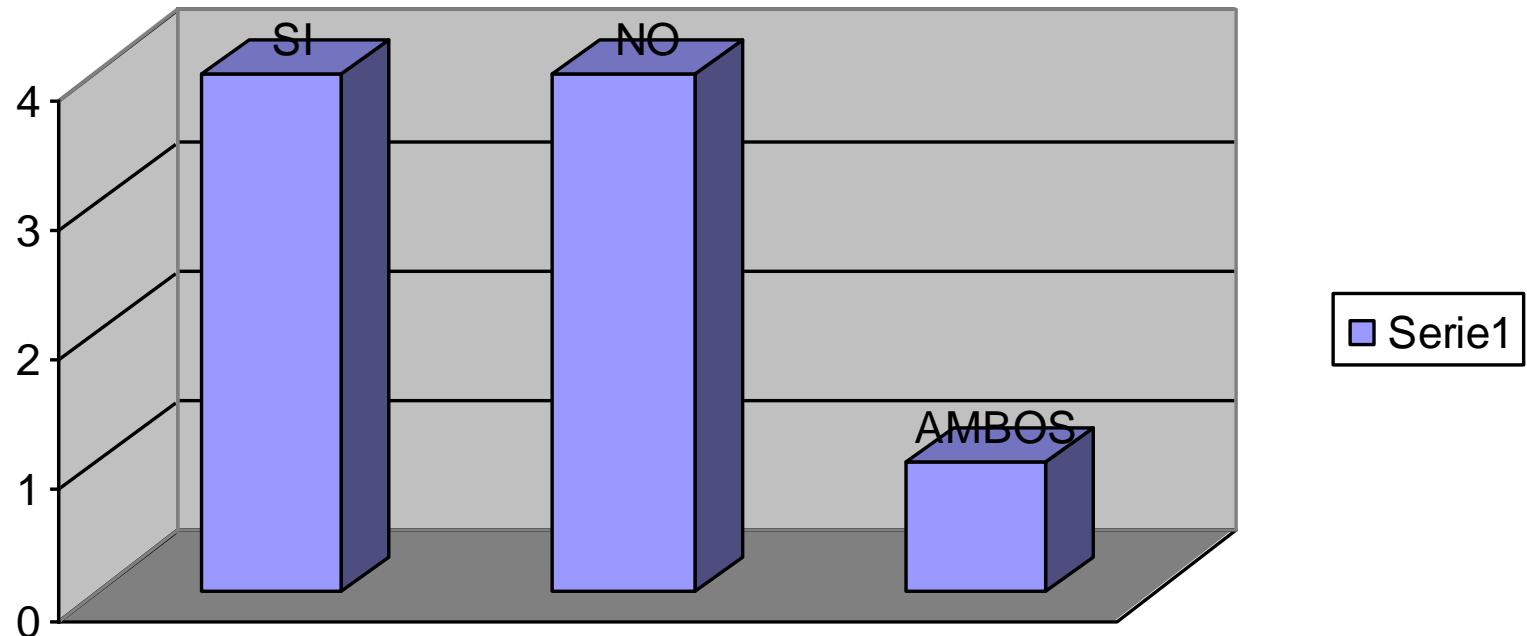
METODOLOGÍA DE SOFTWARE



- MSF (Microsoft Solution Framework)
- XP (eXtreme Programming)
- RUP
- Metodología propia

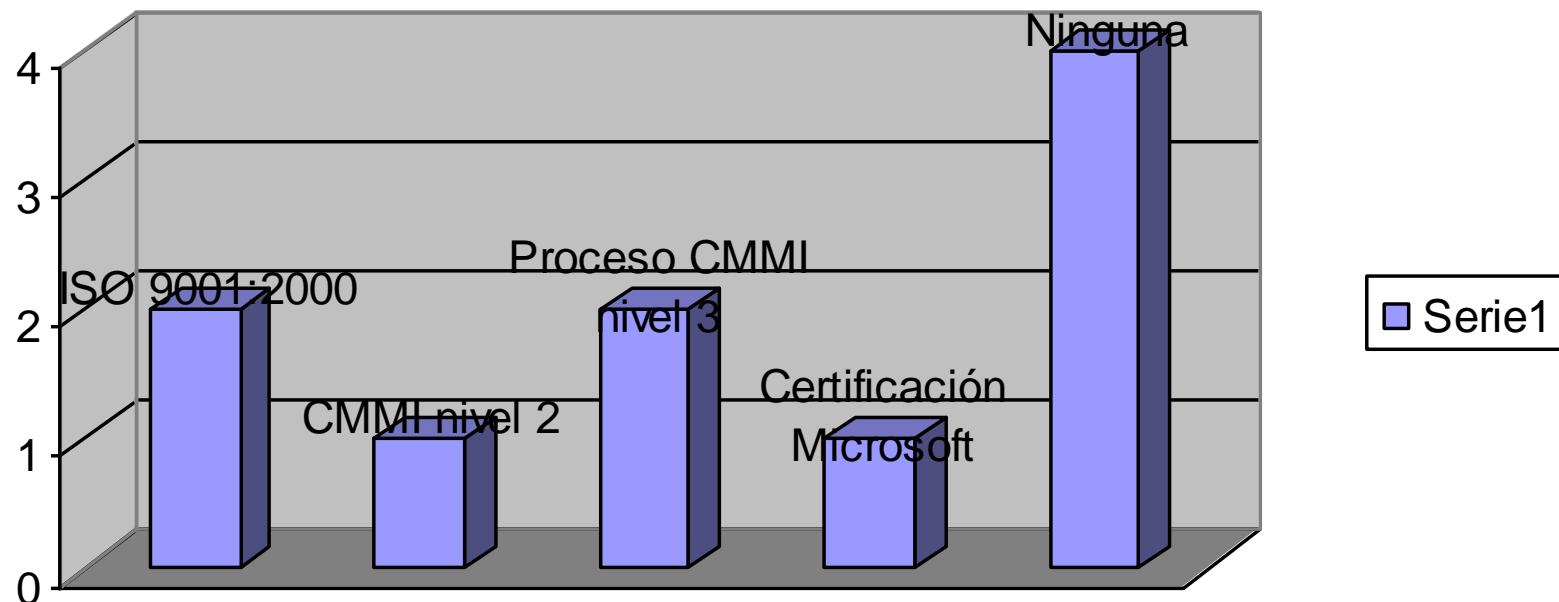
ESTADÍSTICAS

UTILIZACIÓN DE TÉCNICA DE MODELADO PARA DESCRIPCIÓN DE REQUISITOS



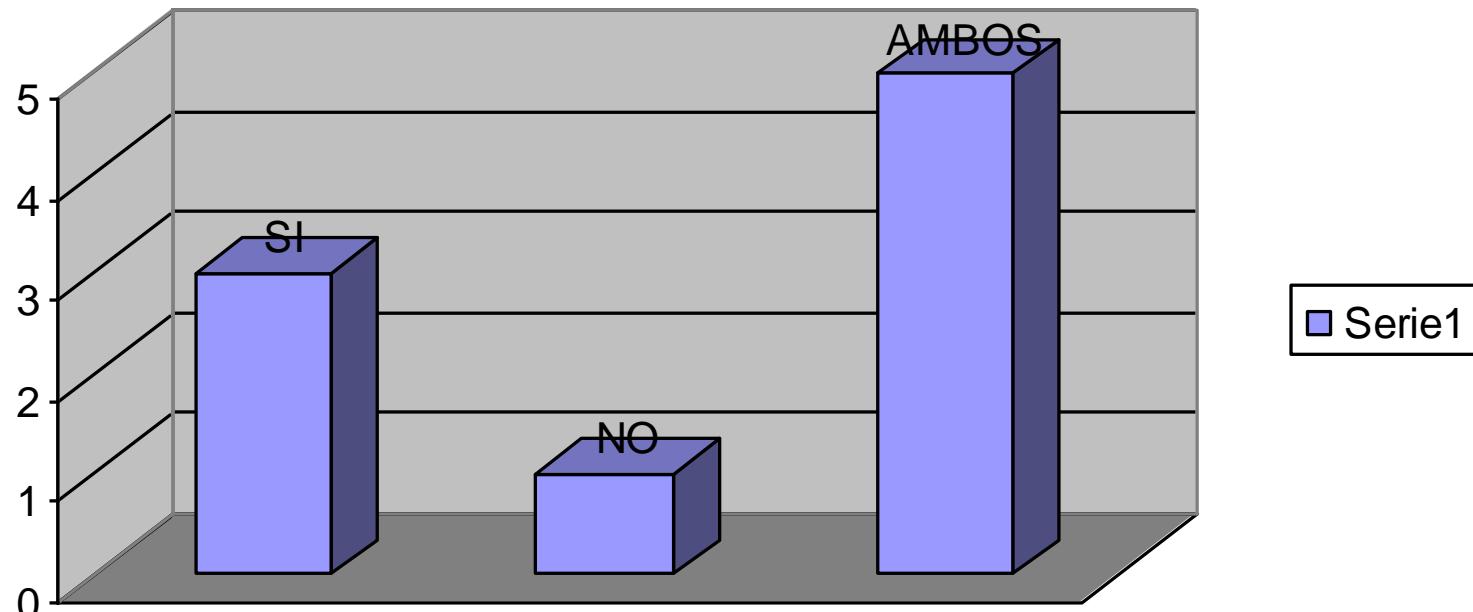
ESTADÍSTICAS

CERTIFICACIÓN EN ESTÁNDARES DE CALIDAD DE SOFTWARE



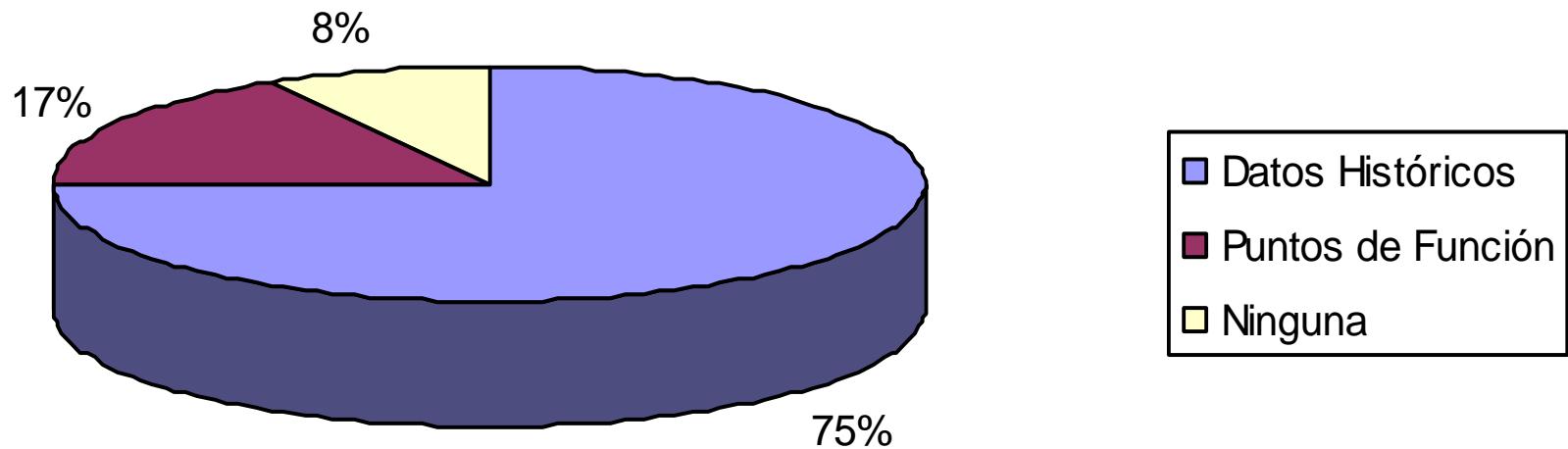
ESTADÍSTICAS

CUMPLIMIENTO DE LA ESTIMACIÓN INICIAL DEL PROYECTO



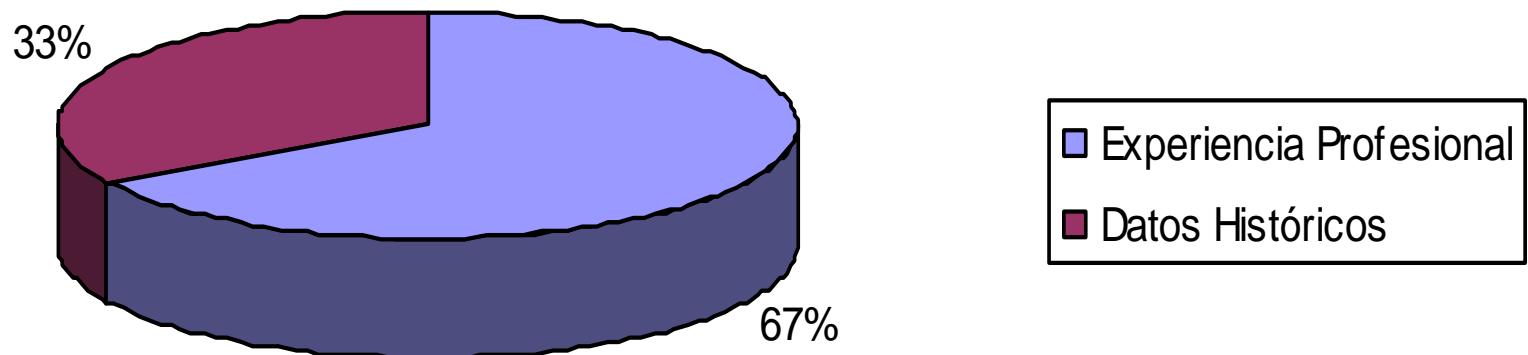
ESTADÍSTICAS

MEDIDA DE TAMAÑO DE PROYECTOS



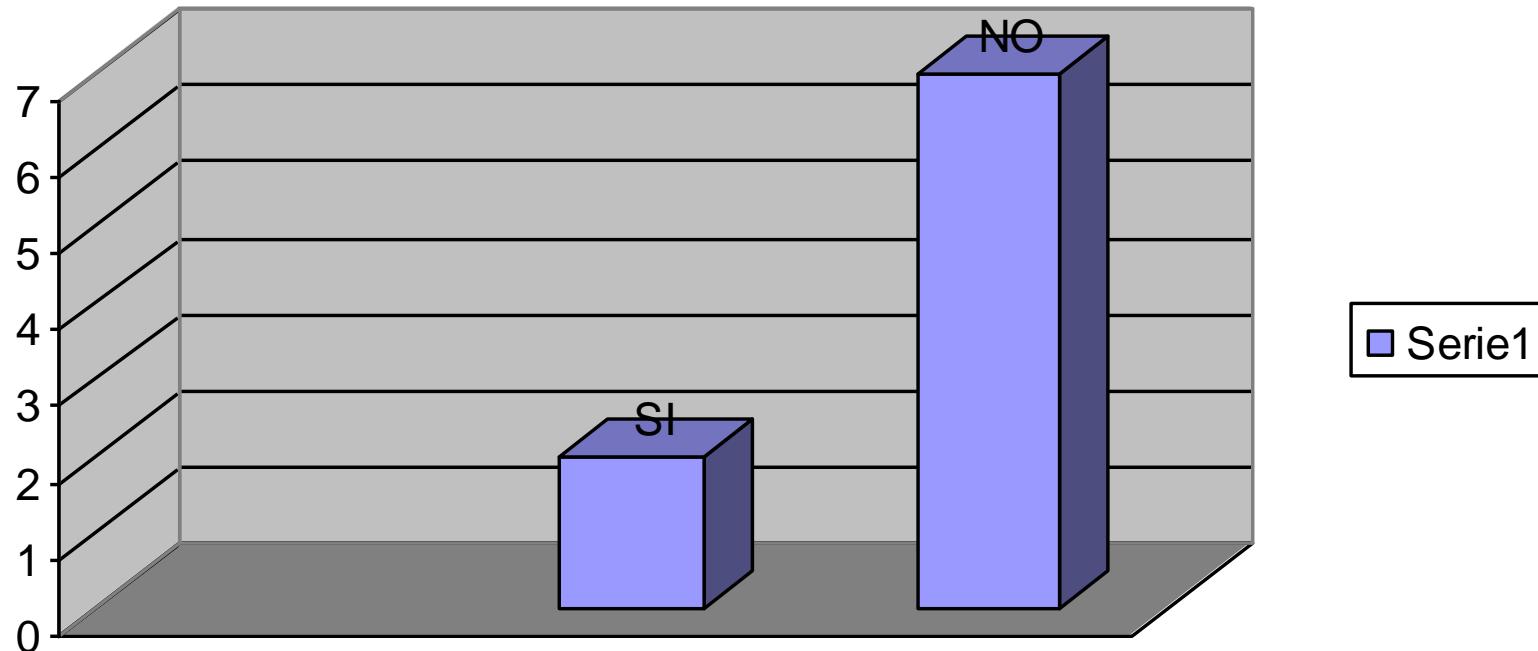
ESTADÍSTICAS

DETERMINACIÓN DE TIEMPO, COSTO Y RRHH ESTIMADO



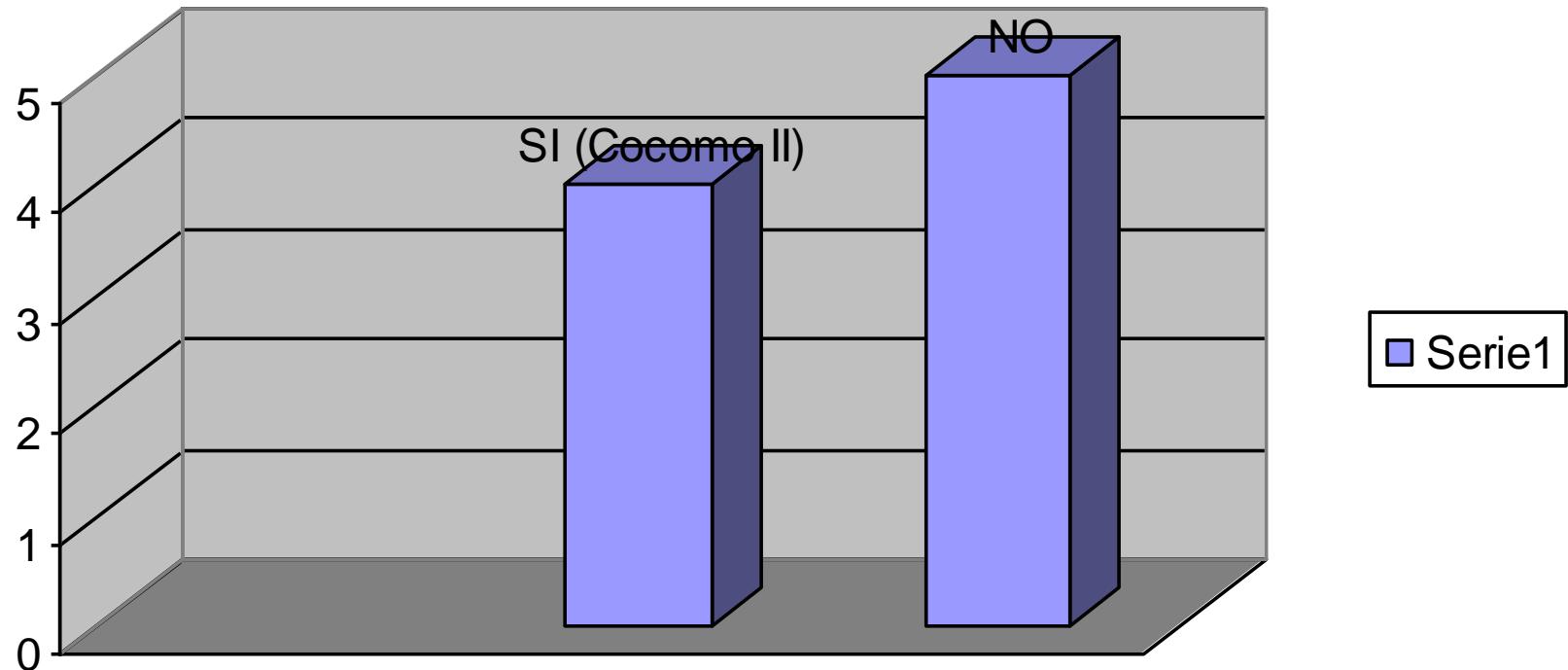
ESTADÍSTICAS

UTILIZACIÓN DE MODELOS CONCEPTUALES PARA ESTIMACIÓN



ESTADÍSTICAS

CONOCIMIENTO DE SISTEMA AUTOMATICO DE PREDICCION DE TIEMPO, COSTO Y RRHH



REFERENCIAS PRINCIPALES

- A Moreno S, Capuchino; Estimación de Proyectos de Software. Master IS. UPM 2005
- Mendenhall, William; Sincich Terry; Probabilidad y Estadística para Ingeniería y Ciencias. Edit. Prentice Hall, 1997, ISBN 968-880-960-8.
- IEEE, ISO; INTERNATIONAL STANDAR ISO/IEC 20926, Software Engineering functional size measurement method; First Ed. 2003.
- Center for Software Engineering, USC; COCOMO II.2000 Model Manual and Software User Manual, 2000. Available at:
<http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/>