



Departamento de Informática  
Universidad de Valladolid  
Campus de Segovia

---

# TEMA 3: ADMINISTRACIÓN DE PROYECTOS

# DEFINICIÓN

- Consiste en organizar, planificar y programar los proyectos de software.
- Aborda actividades que garantizan que el software:
  - Se entregará a tiempo
  - Según el plan preestablecido
  - De acuerdo con los requisitos de la organización.

# ACTIVIDADES

- Redacción de la propuesta
  - Describe los objetivos y como se llevarán a cabo
- Planificación y establecimiento del calendario.
  - Tareas a realizar y plan de trabajo
- Estimación del coste del proyecto
- Supervisión y revisión del proyecto
  - Para comparar los progresos y costes reales con los planeados para hacer ajustes.
- Selección y evaluación del personal
- Redacción y presentación de informes

# PLANIFICACIÓN DEL PROYECTO

- Actividad, dentro de la gestión, que más tiempo consume.
- Es un proceso iterativo que sólo finaliza cuando el proyecto termina.
- El plan debe ser periódicamente revisado con objeto de poder aplicar medidas reguladoras.
- Es imposible planificar sin estimar.

# ESTRUCTURA DE UN PLAN DE PROYECTO SOFTWARE

- En el se fijan los recursos disponibles, se divide el trabajo y se crea un calendario. Incluye:
  - Introducción
    - Definición de objetivos del proyecto y de las restricciones económicas y temporales.
  - Organización del proyecto
    - Organización del equipo: personas involucradas y sus tareas.
  - Análisis de riesgos
    - Posibles riesgos con su probabilidad y propuesta de estrategias de reducción de riesgos.
  - Requisitos de recursos de hardware y software

# ESTRUCTURA DE UN PLAN DE PROYECTO SOFTWARE

- División del trabajo
  - División del proyecto en actividades, marca de hitos y productos a entregar.
- Calendario del proyecto
  - Dependencias entre actividades, tiempo estimado requerido y asignación de personal.
- Mecanismos de supervisión e informes
  - Cuando y que tipos de informes

# ORGANIZACIÓN DE LAS TAREAS

- Las actividades en un proyecto se deben organizar de forma que produzcan salidas tangibles que permitan juzgar el progreso de la actividad.
  - Hitos
  - Productos a entregar

# OBJETIVOS DE UN PLAN DE PROYECTO SOFTWARE

- **Dividir un proyecto en tareas y estimar el tiempo y el coste en recurso de cada tarea.**
- **Organizar las tareas concurrentes para hacer un uso óptimo de los recursos.**
- **Minimizar las dependencias entre tareas para evitar retrasos.**

# PROBLEMAS

- Estimar la dificultad de un problema y por tanto el coste de su solución es difícil.
- La productividad no es proporcional al número de personas que trabajan en una tarea.
- Añadir más personal al final de un proyecto supone siempre problemas debido a la sobrecarga en la comunicación.

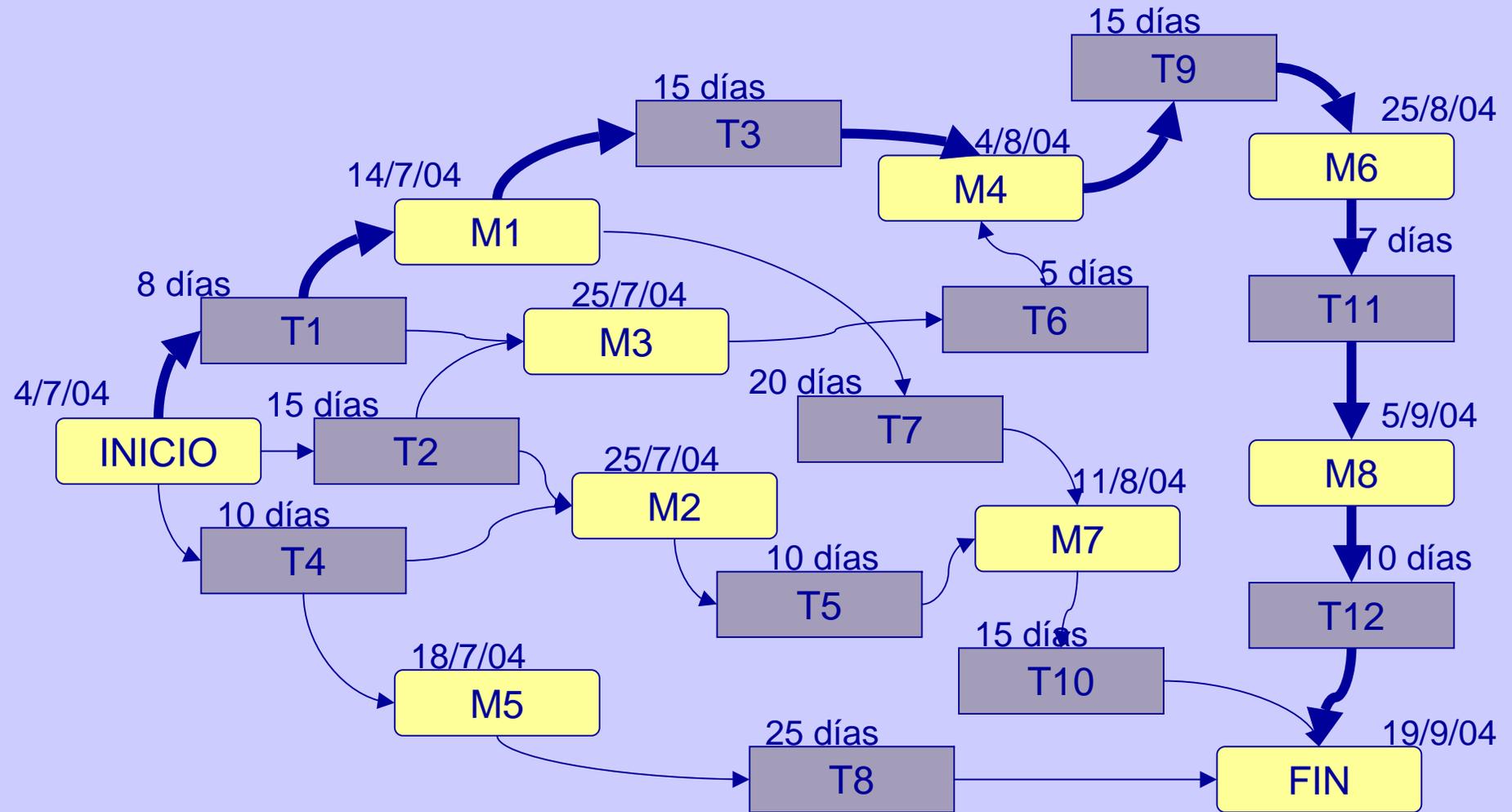
# TÉCNICAS GRÁFICAS PARA ILUSTRAR UNA PLANIFICACIÓN

- Muestran la descomposición en tareas de un proyecto y permiten establecer un calendario.
- Estas tareas no deben ser demasiado pequeñas.
- Diagramas de actividad
  - Muestran las dependencias de las tareas y el camino crítico.
- Diagramas de barras
  - Muestran la planificación sobre el calendario

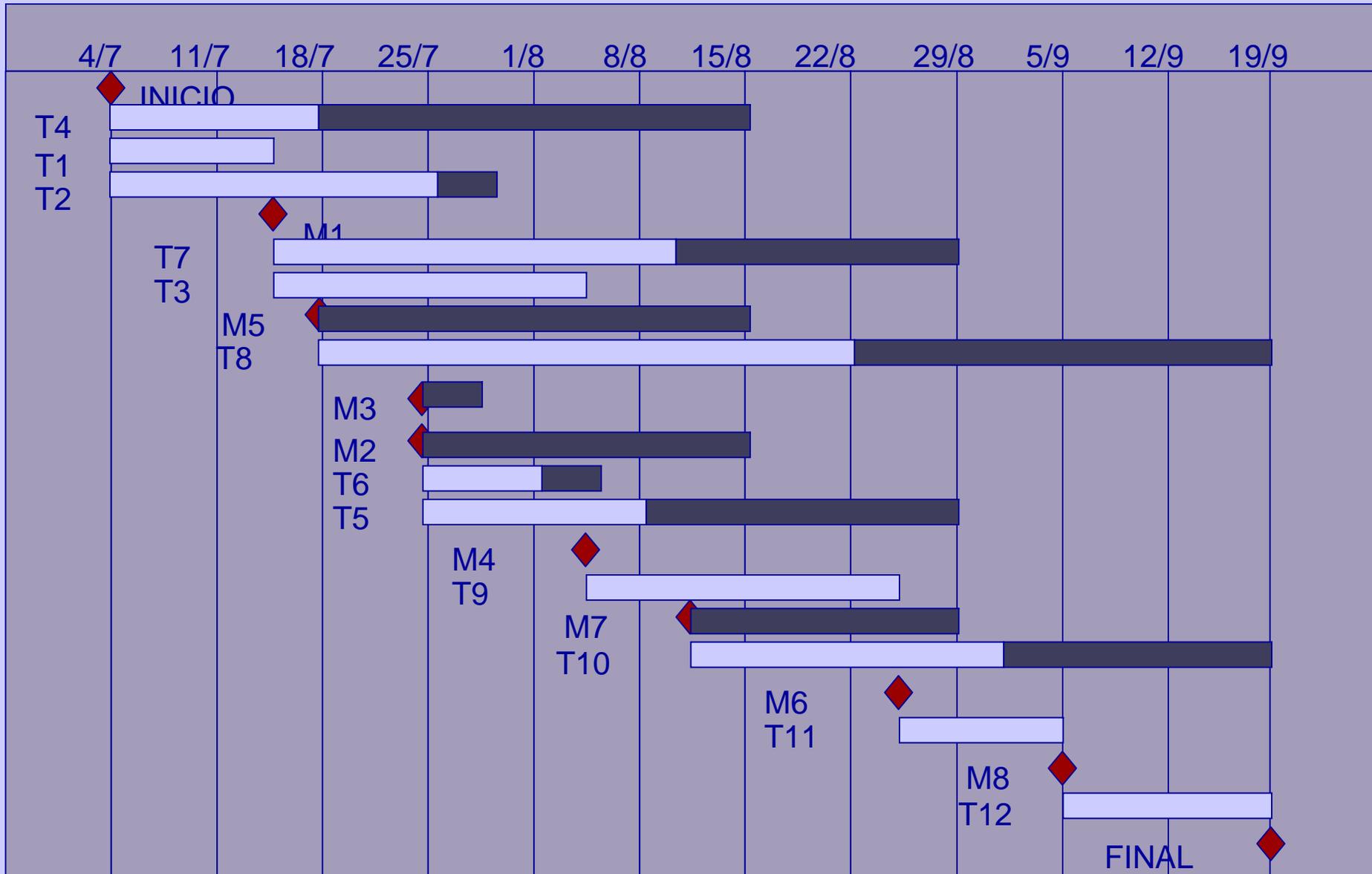
# EJEMPLO DE DIAGRAMA DE ACTIVIDAD

TAREA	DURACIÓN (DÍAS)	DEPENDENCIAS
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2,T4 (M2)
T6	5	T1,T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3,T6 (M4)
T10	15	T5,T7 (M7)
T11	7	T9 (M6)
T12	10	T11(M8)

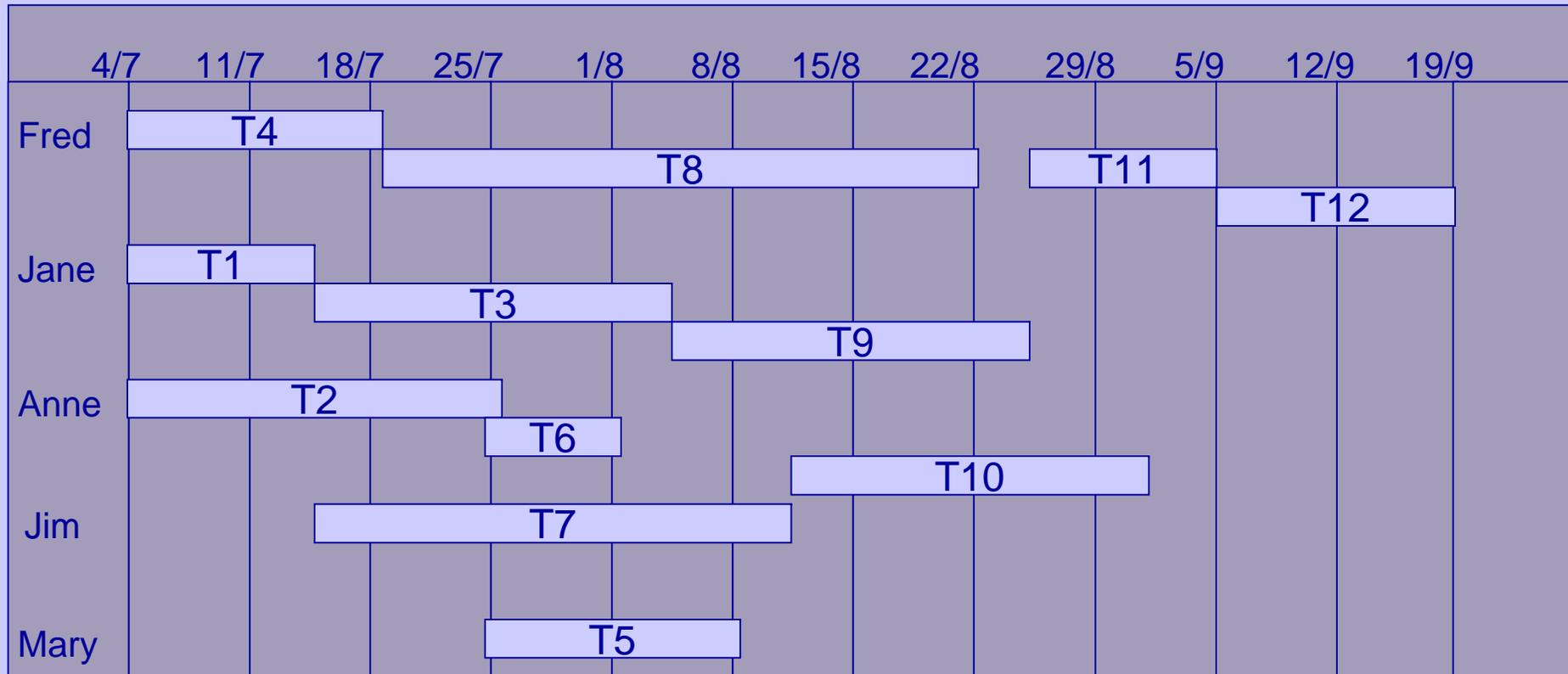
# EJEMPLO DE RED DE ACTIVIDADES



# EJEMPLO DE BARRA DE ACTIVIDADES



# EJEMPLO DE ASIGNACIÓN DE PERSONAL MEDIANTE UN DIAGRAMA DE BARRAS



# PROBLEMA

- En la realización del proyecto “....” se establecen una serie de actividades en las que se detallan las duraciones (en semanas) y las relaciones de precedencia (tabla 1).
  - Representar la red de actividades asociado a este proyecto.
  - Calcular la duración total del proyecto.
  - Determinar el camino crítico y las actividades que lo componen.

# TABLA1: DE DIAGRAMA DE ACTIVIDAD

ACTIVIDAD	ACTIVIDADES PRECEDENTES	DURACIÓN (DÍAS)
T1	T2, T4	3
T2	T3	6
T3	T10	4
T4	T9, T13	8
T5	T9, T13	9
T6	T9, T13	3
T7	T8	5
T8	T10	4
T9	T10	2
T10		6
T11	T6,T7	4
T12	T1,T5,T14	3
T13	T8	7
T14	T6,T7	6

# ESTIMACIÓN DEL COSTE DEL SOFTWARE

- Aborda la predicción de los recursos que se van a requerir para un determinado proceso de desarrollo de software.
- Materias que se estiman:
  - Esfuerzo requerido en recursos humanos, medios materiales, etc..para una actividad
  - Tiempo que puede llevar una actividad.
  - Coste total de una actividad.

# MÉTRICAS DE PRODUCTIVIDAD

- Métricas relacionadas con el tamaño.
  - Basadas en el tamaño de la salida de una actividad del proceso:
    - Número de líneas del código fuente
    - Número de instrucciones del código objeto.
    - Número de páginas de la documentación....
- Métricas relacionadas con la función:
  - Basadas en una estimación de la funcionalidad del software entregado: **puntos de función.**

# ESTIMACIÓN DE COSTES

- No existe una forma simple de hacer una estimación exacta del esfuerzo requerido ya que **Se basan en información poco precisa.**
- Las estimaciones de los costes suelen tender a “autorealizarse”:
  - La estimación determina el presupuesto y el producto se ajusta para que cumpla dicho presupuesto.

# TÉCNICAS DE ESTIMACIÓN DE COSTES

- Modelado algorítmico de costes
  - Se utiliza un análisis previo del coste y otros atributos de otros proyectos realizados.
  - A partir de dicho análisis se obtiene una ecuación matemática que permite predecir los costes.
- Opinión de expertos
  - Se consulta a varios expertos en las técnicas de desarrollo que se van a emplear y en el dominio en el que se va a trabajar. Entre ellos acuerdan una estimación.
- Estimación por analogía
  - Se estima por analogía con otros proyectos ya realizados sobre el mismo dominio de aplicación.

# TÉCNICAS DE ESTIMACIÓN DE COSTES

- Ley de Parkinson
  - El trabajo se extiende para llenar el tiempo disponible
  - El coste se determina en función de los recursos disponibles.
- Precio para ganar
  - Se acuerda la funcionalidad aceptable del sistema en función del coste acordado.

# MODELADO ALGORÍTMICO DE COSTES: COCOMO

- COnstructive COst MOdel
- Se apoya en una estimación previa del tamaño del software en Líneas De Código.
- Las ecuaciones fueron derivadas por Boehm a partir de los datos disponibles de 63 proyectos realizados en la empresa TRW.
- La forma general de las ecuaciones de esfuerzo es:  
$$\text{Esfuerzo} = a \cdot (\text{KLDC})^b$$

# ECUACIONES DE ESFUERZO Y TIEMPO DE COCOMO [Boehm 1981]

Modo de desarrollo	Persona-mes*	Tiempo desarrollo*
Orgánico <sup>1</sup>	PM=3,2 KLDC <sup>1,05</sup>	TD=2,5 PM <sup>0,38</sup>
Empotrado <sup>2</sup>	PM=3,0 KLDC <sup>1,12</sup>	TD=2,5 PM <sup>0,35</sup>
Semi-libre <sup>3</sup>	PM=2,8 KLDC <sup>1,2</sup>	TD=2,5 PM <sup>0,32</sup>

\*valores nominales

- <sup>1</sup>Desarrollado en un entorno estable, con poca innovación técnica con pocas presiones de tiempo y tamaño relativamente pequeño (<50 KLDC).
- <sup>2</sup>Desarrollo de software con requisitos muy restrictivos, con gran volatilidad de requisitos, complejo en un entorno con gran innovación técnica.
- <sup>3</sup>Situaciones intermedias entre los dos modos anteriores

# MODELOS DE APLICACIÓN DE COCOMO

- COCOMO básico:
  - Para estimaciones iniciales moderadamente precisas al comienzo del proyecto cuando no se dispone de detalles (para comenzar a negociar el contrato). Se aplican sólo los valores nominales.
- COCOMO intermedio:
  - Una vez que se han identificado los principales componentes del sistema (cuando se dispone de una especificación de requisitos más o menos terminada). Se calcula a partir del valor nominal aplicando a este la influencia de 15 factores de coste.

# MODELOS DE APLICACIÓN DE COCOMO

- COCOMO detallado:
  - Cuando ya están identificados todos los componentes del sistema ( cuando se dispone de una especificación de requisitos totalmente acabada o el diseño general está bien definido). El modelo dispone de tablas para poder distribuir las cantidades de esfuerzo y de tiempo a lo largo de las distintas fases del proyecto.

# TABLA DE FACTORES DE COSTE DEL MODELO COCOMO

FACTORES	Valor de los factores					
	Muy bajo	Bajo	Medio	alto	Muy alto	extra
Fiabilidad requerida	0,75	0,88	1,00	1,15	1,4	
Tamaño de la base De Datos		0,94	1,00	1,08	1,16	
Complejidad del software	0,70	0,85	1,00	1,15	1,30	1,65
Restricciones de Tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria			1,00	1,06	1,21	1,56
Volatilidad del hardware		0,87	1,00	1,15	1,30	
Restricciones de Tiempo de respuesta		0,87	1,00	1,07	1,15	
Calidad de los analistas	1,46	1,19	1,00	0,86	0,71	
Experiencia con El tipo de aplicación	1,29	1,13	1,00	0,91	0,82	
Experiencia con el hardware	1,21	1,10	1,00	0,90		
Experiencia con el Lenguaje de programación.	1,14	1,07	1,00	0,95		
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Técnicas modernas De programación	1,24	1,10	1,00	0,91	0,82	
Empleo de herramientas	1,24	1,10	1,00	0,91	0,83	
Restricciones a la Duración del proyecto	1,23	1,08	1,00	1,04	1,10	

# EJEMPLO 1 DE APLICACIÓN DE COCOMO

- Estimar el esfuerzo de desarrollo de un sistema de comunicaciones de 30 KLDC de alta complejidad. Afortunadamente se podrá disponer de personal de muy alta cualificación con una gran experiencia específica en este tipo de software. El coste del salario mensual de cada persona es de 1350 euros.

# EJEMPLO DE APLICACIÓN DE COCOMO

- Solución:
  - Esfuerzo nominal= $3,2 \times (30)^{1,05} = 113,79$  personas-mes
  - Esfuerzo= $113,79 \times 1,15(\text{complejidad}) \times 0,7(\text{personal}) \times 0,91(\text{experiencia}) = 83,35$  personas-mes
  - Coste= $83,35 \times 1350 = 112522,5$  euros
  - Tiempo= $2,5 \times (83,35)^{0,38} = 13,42$  meses
  - n° medio de personas= $83,35 / 13,42 = 6,2$  personas
- Si el personal fuera de nivel medio con un salario de 1275 euros:
  - Esfuerzo= $113,79 \times 1,15 \times 0,91 = 119,08$  personas-mes
  - Coste= $119,08 \times 1275 = 151827$  euros

# EJEMPLO 2 DE APLICACIÓN DE COCOMO

- Se desea desarrollar para el avión AIR-96 el software para un controlador de presión del aire en cabina que residente en ROM interactúe sobre los mecanismos de ajuste en función de las señales que le llegan de los sensores, con bastantes cambios en los requisitos debido a las distintas variaciones en el soporte hardware. El tamaño estimado es de 10 KLDC. Consideraremos los factores de costes a nivel nominal excepto la fiabilidad (muy alta), base de datos (reducida), restricciones de memoria, de tiempo de ejecución y de respuesta (muy altas), complejidad (alta), experiencia en el hardware (alta) y gran calidad de los programadores. Estímese el esfuerzo requerido en personas-mes y el tiempo de desarrollo necesario.

# ESTIMACIÓN MEDIANTE PUNTOS DE FUNCIÓN (FPA)

- Este método se basa en una métrica que cuantifica la funcionalidad que se debe entregar al usuario al construir la aplicación [Albrech 1979].

# ESTIMACIÓN MEDIANTE PUNTOS DE FUNCIÓN (FPA)

- Los parámetros que sirven para evaluar dicha funcionalidad son:
  - **número de entradas**
    - Datos que el usuario aporta al sistema (nombres de ficheros, menús de selección).
  - **número de salidas:**
    - Datos que el sistema aporta al usuario (informes, mensajes).
  - **número de ficheros lógicos internos**
    - Ficheros o bases de datos internos al sistema (es decir sólo los utiliza el sistema, ficheros maestros).
  - **número de ficheros externos**
    - Ficheros o bases de datos externos al sistema.(es decir que pueden ser “vistos” por otras aplicaciones).
  - **número de consultas externas.**
    - Entradas que requieren de una respuesta por parte del sistema

# PROCEDIMIENTO: PUNTOS DE FUNCIÓN NO AJUSTADOS

- Los pasos a seguir son:
  - Se debe contar el número de elementos de cada clase
  - Cada elemento de cada clase debe ser clasificado según su grado de complejidad (alta, media o baja).
  - Por último se obtiene los PFNA mediante una suma ponderada de esas cantidades con los pesos que aparecen a continuación

Parámetro significativo	Complejidad baja	Complejidad media	Complejidad alta
Entradas	x3	x4	x6
Salidas	x4	x5	x7
Ficheros internos	x7	x10	x15
Ficheros externos	x5	x7	x10
Consultas externas	x3	x4	x6

# CRITERIOS PARA EVALUAR LA COMPLEJIDAD DE LOS ELEMENTOS DE CÁLCULO

Ficheros lógicos externos e internos				Salidas y consultas				Entradas			
Registros elementales	Datos elementales			Tipos de ficheros	Datos elementales			Tipos de ficheros	Datos elementales		
	1-9	20-50	>51		1-5	6-19	>20		1-4	5-15	>16
1	Baja	Baja	Media	0-1	Baja	Baja	Media	0-1	Baja	Baja	Media
2-5	Baja	Media	Alta	2-3	Baja	Media	Alta	2-3	Baja	Media	Alta
>6	Media	Alta	Alta	>4	Media	Alta	Alta	>3	Media	Alta	Alta

# EJEMPLO

- Se dispone de una aplicación que comprueba la corrección ortográfica a partir de un texto. El corrector acepta como entradas el fichero que contiene el documento y un fichero con un diccionario personal opcional. La no presencia del fichero con el documento a corregir da lugar a un mensaje de error. El corrector lista todas aquellas palabras que no se encuentran tanto en el diccionario personal como en el diccionario interno de la aplicación correctora. El usuario puede finalmente consultar el número de palabras procesadas y el número de errores que se encontraron durante el proceso de corrección. Para el cálculo se supondrá que la complejidad de todos los factores es media.

# EJEMPLO

## Entradas: 2

- nombre del fichero que contiene el a corregir documento
- nombre del fichero con el diccionario personal

## Salidas: 3

- informe sobre errores ortográficos
- informe sobre palabras procesadas
- mensaje de errores

## Ficheros externos: 2

- fichero con el documento
- fichero con el diccionario personal

## Ficheros internos: 1

- fichero con el diccionario interno

## Consultas externas: 2

- palabras procesadas
- errores ortográficos

# EJEMPLO

$$\text{PFNA} = (2 \times 4)_i + (3 \times 5)_o + (2 \times 7)_{of} + (1 \times 10)_{if} + (2 \times 4)_{ei}$$

$$\text{PFNA} = 55$$

# AJUSTE DE LOS PFNA

- Una vez obtenidos los PFNA deben ser ajustados mediante un Factor de Ajuste (FA).
- Existen 14 factores que contribuyen a la complejidad de una aplicación, cada uno de ellos valorados dentro de una escala de 0 a 5.
- FA se obtiene a partir de la suma de los 14 factores de complejidad (FC) mediante la siguiente ecuación:

$$FA=(0.01 \times \sum FC)+0.65$$

- Los puntos de función ajustados (PF) se obtienen de la siguiente forma:

$$PF=PFNA \times FA$$

# FACTORES DE COMPLEJIDAD (FC)

Factores de Complejidad (FC)	0-5	Factores de Complejidad	0-5
Comunicación de datos		Funciones distribuidas	
Rendimiento		Gran carga de trabajo	
Frecuencia de transacciones		Entrada on-line de datos	
Requisitos de manejo del usuario final		Actualizaciones on-line	
Procesos complejos		Utilización con otros sistemas	
Facilidad de mantenimiento		Facilidad de operación	
Instalación en múltiples lugares		Facilidad de cambio	

## Significado de los factores de complejidad

# FACTORES DE COMPLEJIDAD PARA EL EJEMPLO

Factores de Complejidad (FC)	0-5	Factores de Complejidad (FC)	0-5
Comunicación de datos	3	Funciones distribuidas	0
Rendimiento	5	Gran carga de trabajo	0
Frecuencia de transacciones	0	Entrada on-line de datos	3
Requisitos de manejo del usuario final	3	Actualizaciones on-line	3
Procesos complejos	5	Utilización con otros sistemas	0
Facilidad de mantenimiento	0	Facilidad de operación	3
Instalación en múltiples lugares	0	Facilidad de cambio	3

# EJEMPLO

$$FA = 0.65 + (0.01 \times 28) =$$

$$PF = FC \times PFNA = 0.93 \times 55 = 51,15$$

- Si 1 punto de función equivale a 128 líneas de código en C, entonces:
  - Líneas en código de C:  $51,15 \times 128 = 6547$  LDC

# ESTRATEGIA DE LA ESTIMACIÓN DE COSTES

- La estimación se debería realizar aplicando varias técnicas.
- Si no se obtiene resultados aproximados a través de las distintas técnicas es que se dispone de poca información.
  - Solución: buscar más información hasta que se consiga que todas las técnicas empleadas converjan a un valor.
- “Precio para ganar” es en algunos casos la única técnica aplicable.

# ADMINISTRACIÓN DE RIESGOS

- Consiste:
  - En anticipar los riesgos que pueden poner en peligro el proyecto
  - Emprender acciones para evitar dichos riesgos.
- Un riesgo es:
  - una probabilidad de que una circunstancia adversa ocurra.
- Categorías de riesgos:
  - **Riesgos del proyecto:** afectan al calendario o a los recursos del proyecto.
  - **Riesgos del producto:** afectan a la calidad o desempeño del software en desarrollo.
  - **Riesgos del negocio:** afectan a la organización que desarrolla el software.

# PROCESO DE ADMINISTRACIÓN DE RIESGOS

- **Identificación de riesgos:** se identifican los posibles riesgos del proyecto, del producto y del negocio.
- **Análisis de riesgos:** Se valoran las probabilidades y consecuencias de estos riesgos.
- **Planificación de riesgos:** se crean planes para abordar los riesgos, ya sea para evitarlos o para minimizar sus efectos sobre el proyecto.
- **Supervisión de riesgos:** Se valoran los riesgos de forma constante y se revisan los planes para la mitigación de riesgos tan pronto está disponible la información sobre dichos riesgos.

# FACTORES DE RIESGO

TIPO DE RIESGOS	INDICADORES POTENCIALES
Tecnología	Entrega retrasada del hardware o de la ayuda al software
Personas	Baja moral del personal, malas relaciones entre los miembros del equipo, disponibilidad de empleo
Organización	Mala coordinación, falta de acciones por parte del administrador principal
Herramientas	Rechazo de los miembros del equipo a utilizar herramientas, quejas sobre las herramientas CASE, petición de estaciones más potentes.
Requerimientos	Peticiones de muchos cambios en los requerimientos, quejas del cliente
Estimación	Fracaso en el cumplimiento de los tiempos acordados y en la eliminación de defectos detectados.

# ALGUNOS DE LOS POSIBLES RIESGOS DURANTE EL DESARROLLO DE SOFTWARE

RIESGO	TIPO DE RIESGO	DESCRIPCIÓN
Rotación de personal	Proyecto	Personal con experiencia abandona el proyecto antes de que finalice
Cambio de administración	Proyecto	Se puede dar un cambio de administración organizacional con diferentes prioridades
No disponibilidad de software	Proyecto	El hardware esencial para el proyecto no será entregado a tiempo.
Cambio de requerimientos	Proyecto y Producto	Puede haber más cambios en los requerimientos que los anticipados.
Retrasos en la especificación	Proyecto y Producto	Las especificaciones de las interfaces esenciales no estarán a tiempo
Subestimación del tamaño	Proyecto y Producto	El tamaño del sistema ha sido subestimado
Bajo desempeño de las herramientas CASE	Producto	Las herramientas CASE que ayudan al proyecto no tienen el desempeño anticipado

**THAT'S ALL FOLKS**

# SIGNIFICADO DE LOS FC

- 1. Comunicación de Datos: los datos o información de control que la aplicación utiliza se envía o recibe a través de las facilidades de comunicación.
  - 0 Aplicación es batch exclusivamente
  - 1-2 Impresión o entrada de datos remota
  - 3-5 Teleproceso (TP) interactivo
  - 3 TP interface a un proceso batch
  - 5 La aplicación es interactiva predominantemente
- 2. Función Distribuída. "Distribuída" significa que los componentes (o los datos) de la aplicación están distribuídos en dos o más procesadores diferentes (esto también incrementa el factor anterior).
  - 0 La aplicación no ayuda a la transferencia de datos o a la función de procesamiento entre los componentes del sistema
  - 1 La aplicación prepara datos para el usuario final de otro procesador
  - 2-4 Los datos se preparan para transferencia, se transfieren y se procesan en otro componente del sistema
  - 5 Las funciones de procesamiento se realizan dinámicamente en el componente más apropiado del sistema.

- 3. Rendimiento: referido a la importancia de respuesta dentro de todo el sistema
  - 0-3 Análisis y diseño de las consideraciones del rendimiento son estándar. No se precisan requerimientos especiales por parte del usuario
  - 4 En la fase de diseño se incluyen tareas del análisis del rendimiento para cumplir los requerimientos del usuario
  - 5 Además se utilizan herramientas de análisis del rendimiento en el diseño, desarrollo e instalación
- 4. Configuración utilizada masivamente: referente a la importancia del entorno. Esto es, si hay restricciones de memoria o del hardware.
  - 0-3 La aplicación corre en una máquina estándar sin restricciones de operación
  - 4 Restricciones de operación requieren características específicas de la aplicación en el procesador central
  - 5 Además hay restricciones específicas a la aplicación en los componentes distribuidos del sistema.
- 5. Tasas de Transacción: una alta llegada de transacciones provoca problemas más allá de los de la característica 3
  - 0-3 Las tasas son tales que las consideraciones de análisis de rendimiento son estándares
  - 4 En la fase de diseño se incluyen tareas de análisis de rendimiento para verificar las altas tasas de transacciones
  - 5 Además se utilizan herramientas de análisis del rendimiento.

- 6. Entrada On-Line de datos
  - 0-2 Hasta el 15% de las transacciones tienen entrada interactiva
  - 3-4 15% al 30% tienen entrada interactiva
  - 5 30% al 50% tienen entrada interactiva.
- 7. Diseño para la eficiencia de usuario final
  - 0-3 No se especifican requerimientos especiales
  - 4 Se incluyen tareas de diseño para la consideración de factores humanos
  - 5 Además se utilizan herramientas especiales o de prototipado para promover la eficiencia.
- 8. Actualización On-Line
  - 0 Nada
  - 1-2 Actualización on line de los ficheros de control. El volumen de actualización es bajo y la recuperación fácil.
  - 3 Actualización on line de la mayoría de los ficheros internos lógicos
  - 4 Además es esencial la protección contra la pérdida de datos
  - 5 Además se considera el coste de recuperación de volúmenes elevados.

- 9. Complejidad del procesamiento: esto es, complejidad interna más allá de la media en lo referente a la entrada, salida o lógica de procesamiento ¿Qué características tiene la aplicación? mucho procesamiento matemático y/o lógico • procesamiento complejo de las entradas procesamiento complejo de las salidas muchas excepciones de procesamiento, muchas transacciones incompletas y mucho reprocesamiento de las transacciones procesamiento de seguridad y/o control sensitivo
  - 0 No se aplica nada de esto
  - 1 Se aplica alguna cosa
  - 2 Se aplican dos cosas
  - 3 Se aplican tres cosas
  - 4 Se aplican cuatro cosas
  - 5 Se aplica todo.

- 10. Utilizable en otras aplicaciones: el código se diseña para que sea compartido o utilizable por otras aplicaciones (no confundir con 13).
  - 0-1 Una aplicación local que responde a las necesidades de una organización usuaria
  - 2-3 La aplicación utiliza o produce módulos comunes que consideran más necesidades que las del usuario
  - 4-5 Además, la aplicación se "empaquetó" y documentó con el propósito de fácil reutilización
- 11. Facilidad de Instalación
  - 0-1 No se requieren por parte del usuario facilidades especiales de conversión e instalación
  - 2-3 Los requerimientos de conversión e instalación fueron descritos por el usuario y se proporcionaron guías de conversión e instalación
  - 4-5 Además se proporcionaron y probaron herramientas de conversión e instalación
- 12. Facilidad de Operación
  - 0 No se especifican por parte del usuario consideraciones específicas de operación
  - 1-2 Se requieren, proporcionan y prueban procesos específicos de arranque, backup y recuperación
  - 3-4 Además la aplicación minimiza la necesidad de actividades manuales, tales como instalación de cintas y papel
  - 5 La aplicación se diseña para operación sin atención

- 13. Puestos Múltiples.
  - 0 El usuario no requiere la consideración de más de un puesto
  - 1-3 Se incluyeron necesidades de varios puestos en el diseño
  - 4-5 Se proporciona documentación y plan de apoyo para soportar la aplicación en varios lugares
- 14. Facilidad de Cambio: esfuerzo específico de diseño para facilitar cambios futuros.
  - 0 No hay requerimientos especiales del usuario para minimizar o facilitar el cambio
  - 1-3 Se proporciona capacidad de consulta flexible
  - 4-5 Datos importantes de control se mantienen en tablas que son actualizadas por el usuario a través de procesos on-line interactivos

[volver](#)