Ingeniería de software II

Gestión del Proyecto
Planificación temporal

2025

Planificación Temporal

Ingeniería de Software II – 2025

Planificación Temporal

Es una actividad que distribuye el esfuerzo estimado a lo largo de la duración prevista del proyecto

La precisión de la planificación temporal es muy importante para no generar clientes insatisfechos, costos adicionales, reducción del impacto en el mercado, etc.

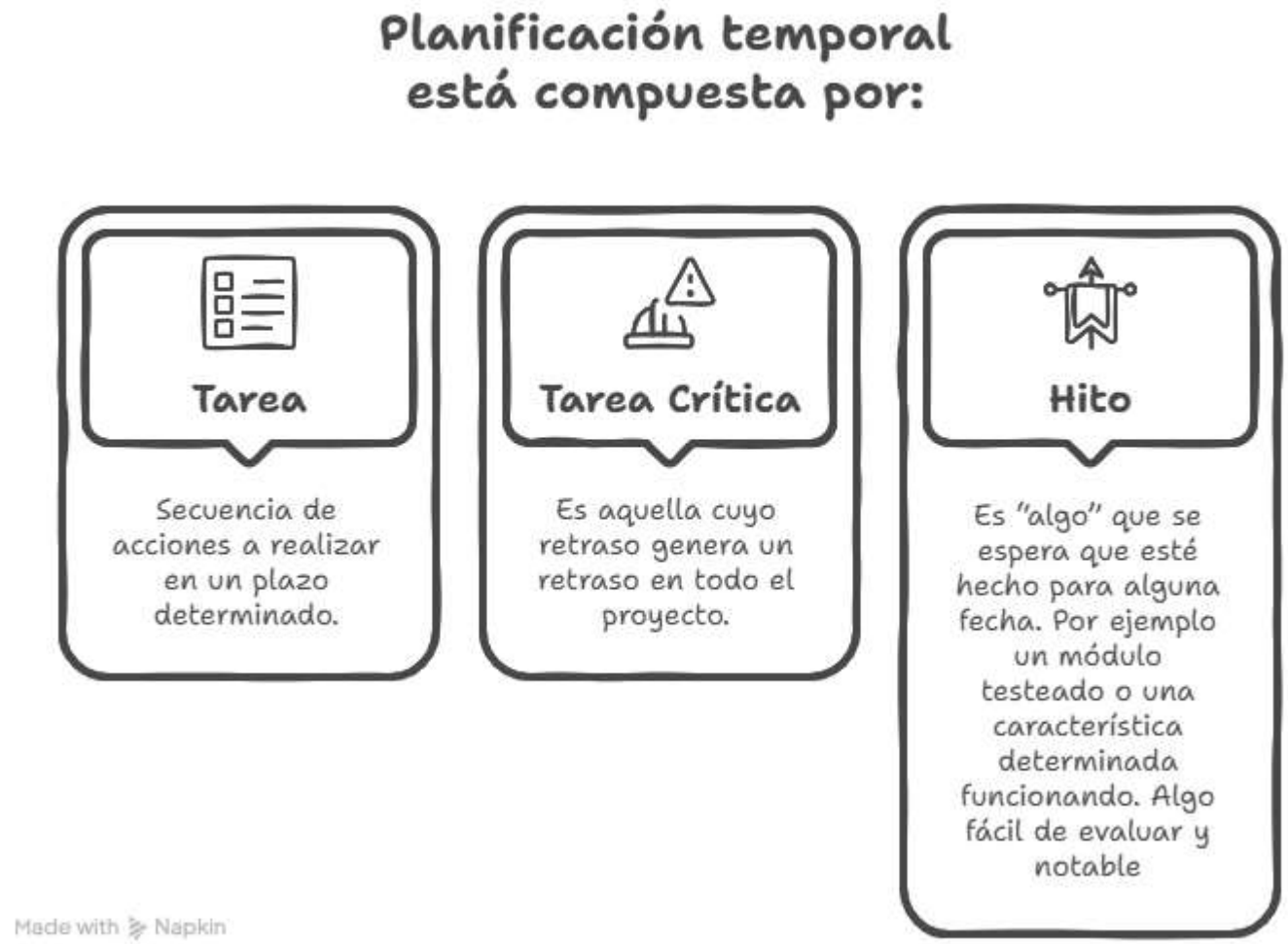
Planificación Temporal

- La ***calendarización del proyecto*** de software es una acción que distribuye el esfuerzo estimado a través de la duración planificada del proyecto, asignando el esfuerzo a ***tareas*** específicas del desarrollo de software.

Calendarización

» Las fechas de los proyectos pueden ser de dos tipos:

Planificación Temporal-Calendarización



Planificación Temporal

Fuente: Pfleeger Cap. 3

Planificación Temporal-Calendarización

Red de tareas

- Es una representación gráfica del flujo de las tareas desde el inicio hasta el fin de un proyecto
- En algunos casos los conjuntos de tareas permiten realizar algunas actividades en paralelo.
- Representan la secuencia de las tareas y su interdependencia

Fuente: Pressman Cap. 7

Planificación Temporal-Calendarización

El conjunto de tareas variará dependiendo del tipo de proyecto y grado de rigor.
Los factores que influyen en el conjunto de tareas a elegir son :

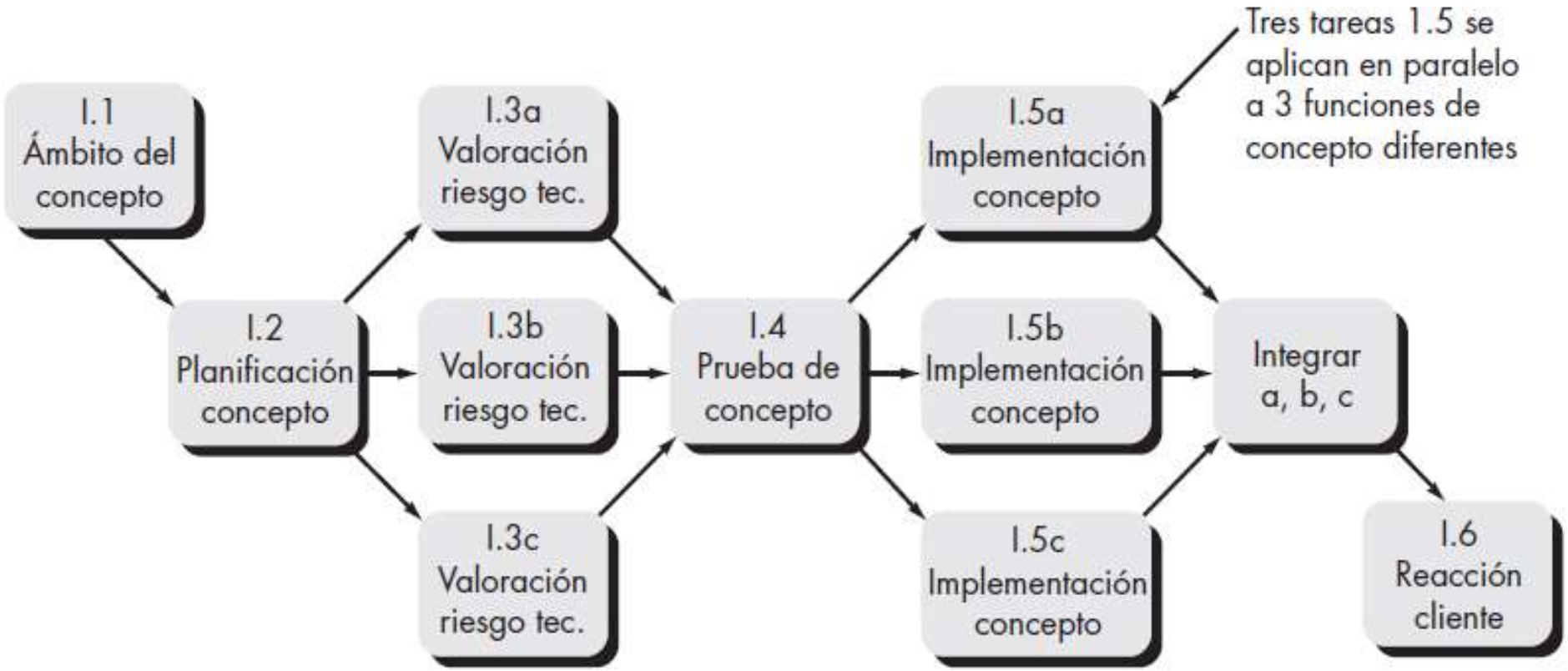
- Tamaño del proyecto
- Número de usuarios potenciales
- Criticidad del proyecto
- Estabilidad de los requerimientos
- Facilidad de comunicación con el cliente/usuario
- Madurez de la tecnología aplicable
- Restricciones

entre otros

Fuente: Pressman Cap. 7

Planificación Temporal

Red de tareas



Fuente:

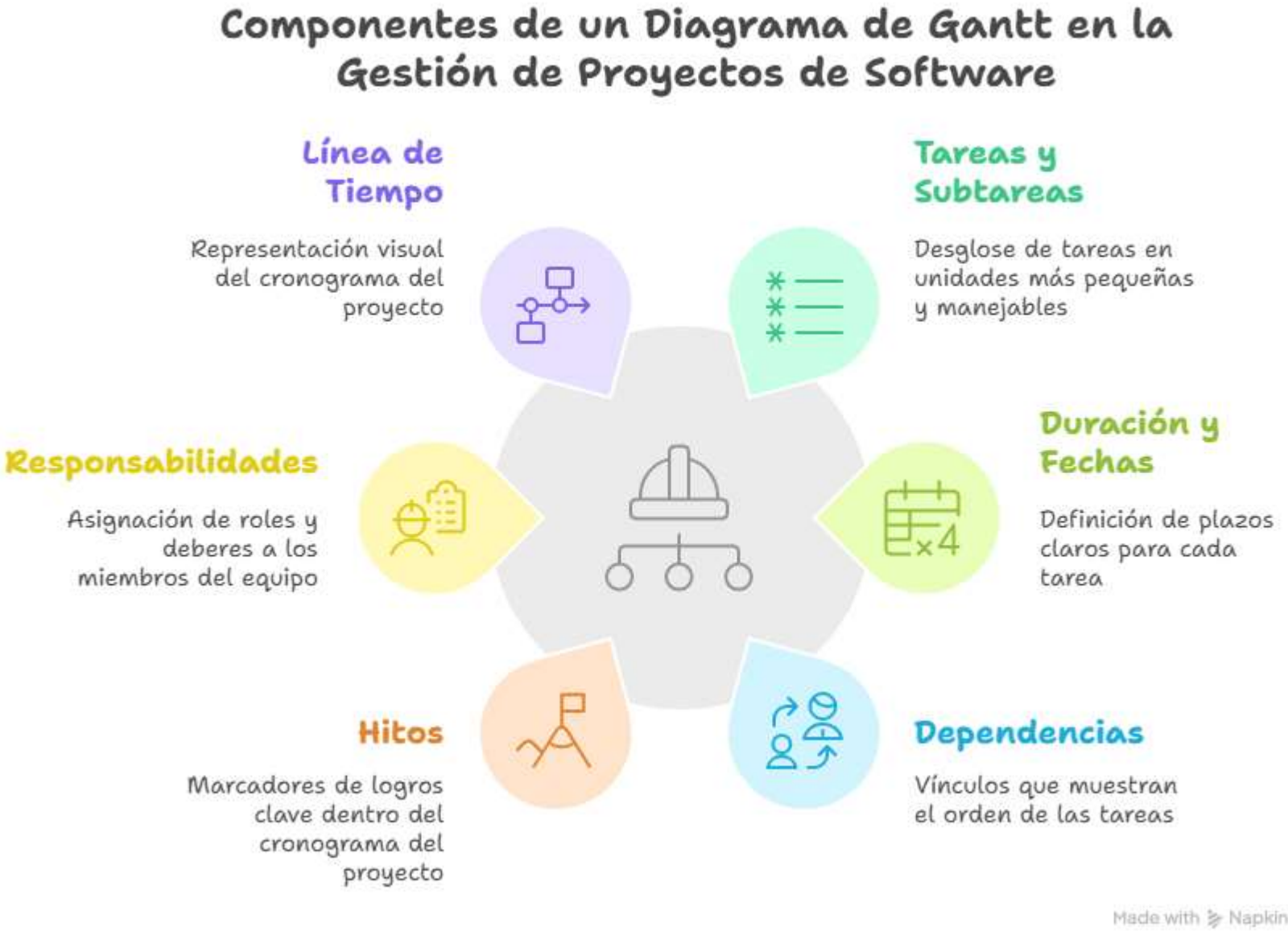
Método de planificación temporal
GANTT

Un diagrama de Gantt es una herramienta visual de gestión de proyectos utilizada para planificar y rastrear el progreso de diversas tareas y actividades dentro del ciclo de vida de un proyecto de software.

Funciona como una línea de tiempo visual, ofreciendo una visión general de alto nivel de los cronogramas del proyecto, lo que simplifica la gestión de planes complejos que involucran múltiples equipos y plazos cambiantes.

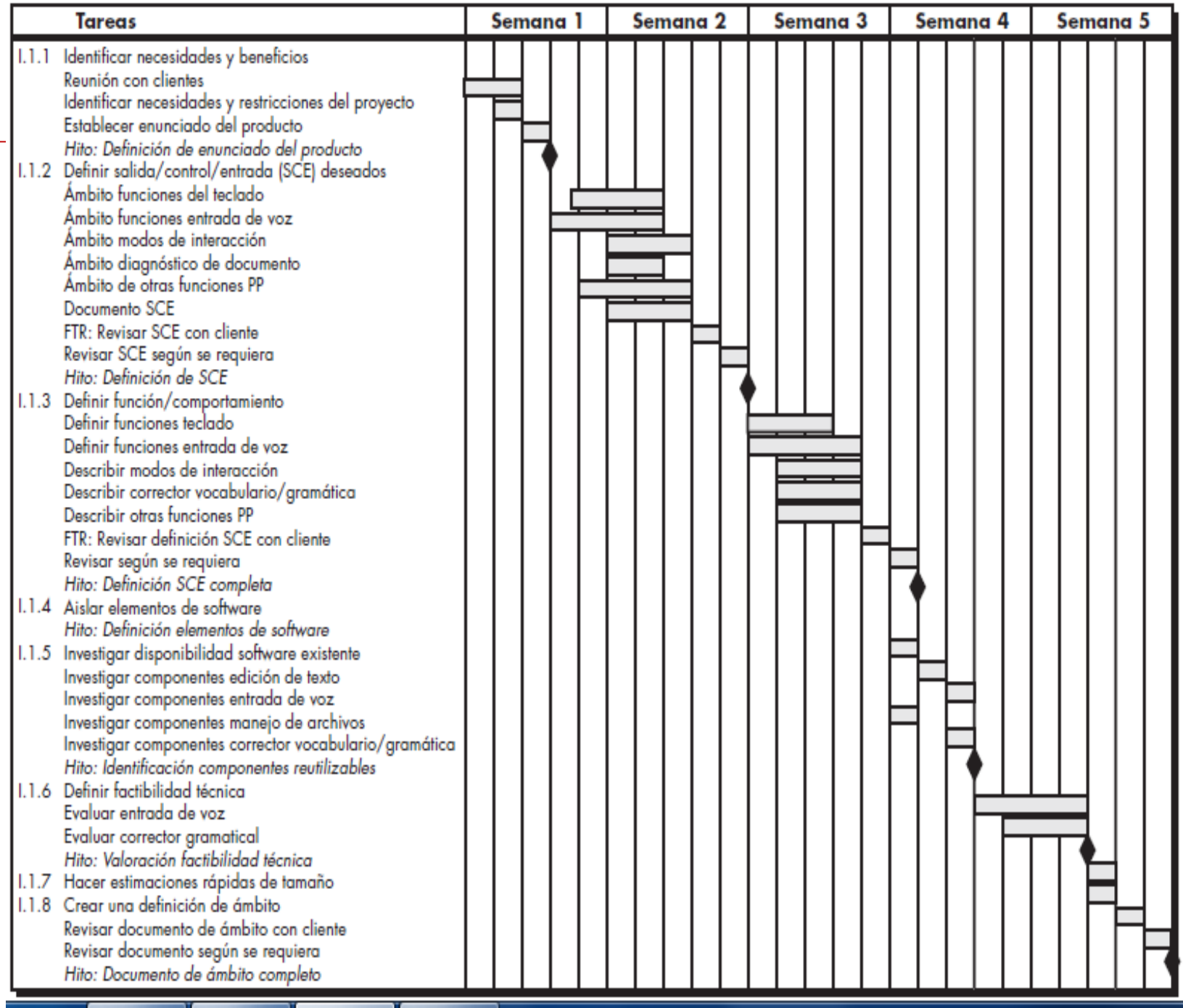
Fuente: Pfleeger Cap 3

Método de planificación temporal
GANTT



Representación gráfica

Diagrama de GANTT



PERT y CPM

PERT, o Técnica de Evaluación y Revisión de Programas, es una herramienta de gestión de proyectos utilizada para analizar las estimaciones de tareas en un cronograma y evaluar las opciones de la ruta crítica, especialmente cuando las duraciones de las tareas son inciertas.

Por otro lado, **CPM,** o Método de la Ruta Crítica, es un método de gestión de proyectos que identifica la secuencia de actividades que determinan el tiempo mínimo de finalización de un proyecto, centrándose en tareas bien definidas con duraciones conocidas. Ambos métodos ayudan a desglosar proyectos complejos en tareas manejables, visualizar cronogramas, identificar dependencias y, en última instancia, apuntan a la finalización del proyecto a tiempo.

Fuente:

Método de planificación temporal

PERT (Program Evaluation & Review Technique):

- Creado para proyectos del programa de defensa del gobierno norteamericano entre 1958 y 1959.
- Se utiliza para controlar la ejecución de proyectos con gran número de actividades que implican investigación, desarrollo y pruebas.
- Red de tareas con Fechas tempranas, tardías, Camino crítico
- Probabilístico

Fuente:

Método de planificación temporal

CPM (Critical Path Method):

- Desarrollado para dos empresas americanas entre 1956 y 1958.
- Se utiliza en proyectos en los que hay poca incertidumbre en las estimaciones.
- Tiempo de inicio temprano y tardío
- Determinístico

Fuente:

Método de planificación temporal

- PERT y CPM

  Actualmente se ha tomado lo mejor de ambos
  vuelto uno solo, conocido como *Método del Co*

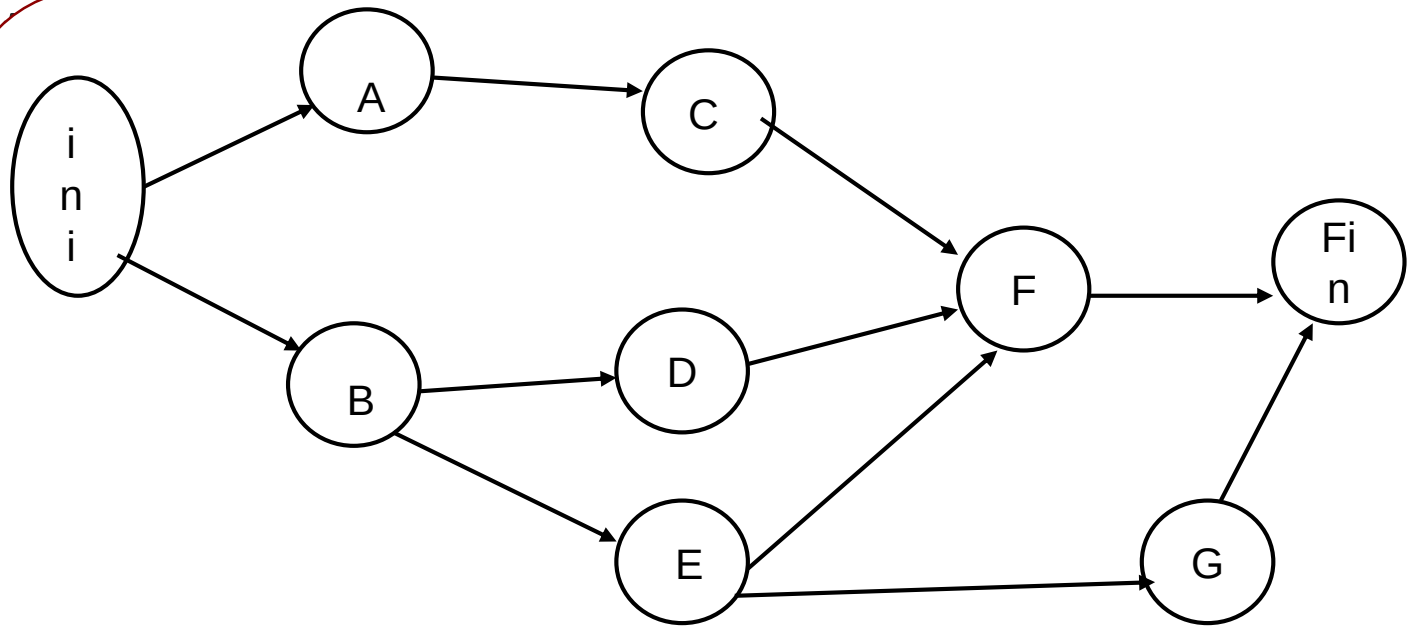
1. *Establecer lista de tareas*
2. *Fijar dependencia en*
3. *Construir la red*
4. *Numer*
5. *Calcular la fech*

   Tei = Fecha tempr

   Tai = Fecha tardía
6. *Calcular el camino*

   ==> Tei = Tai

Tarea	Precedida por	Duración
A	-	2
B	-	5
C	A	4
D	B	7



Método de planificación temporal
PERT - CPM

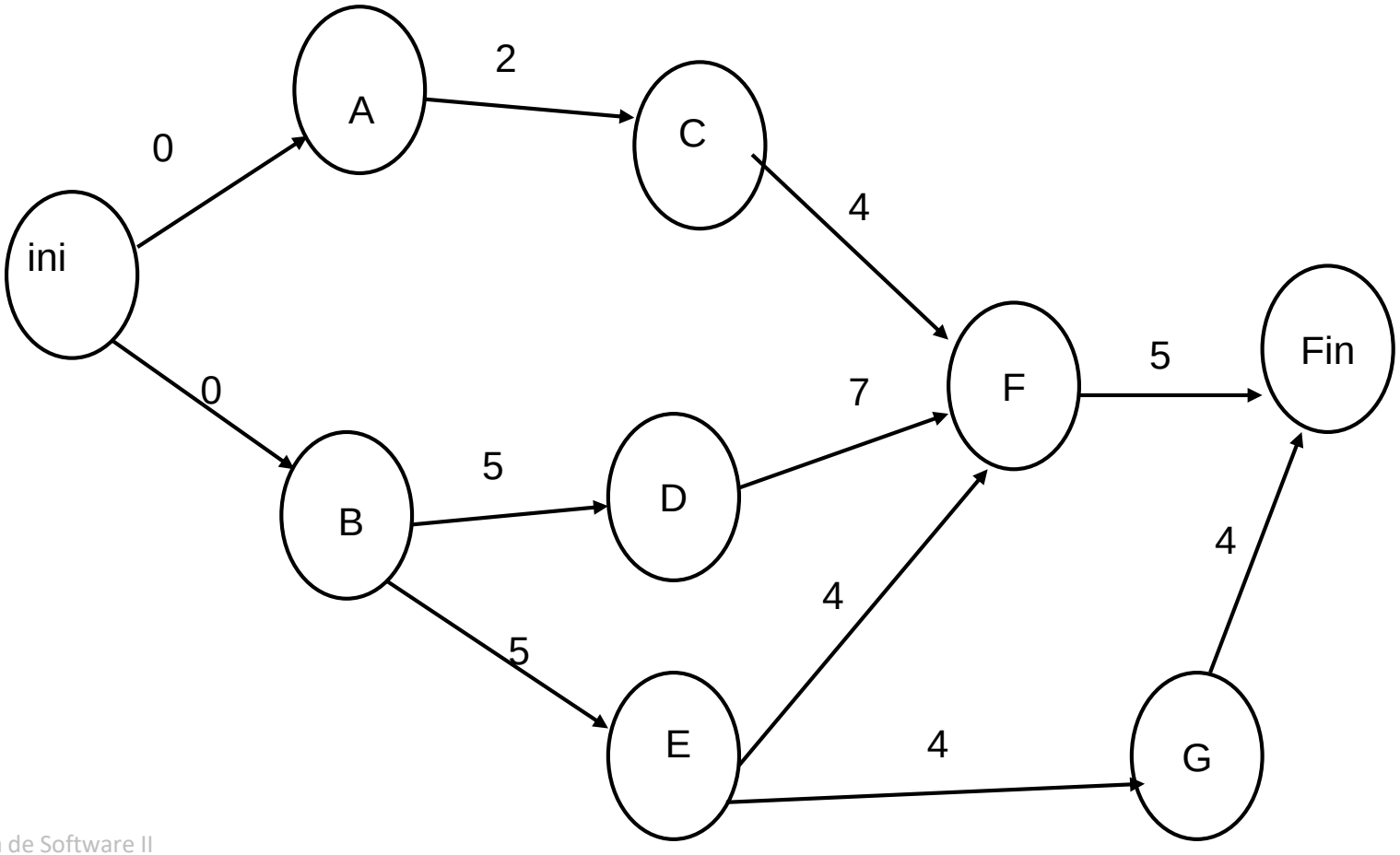
Ejemplo

1. *Establecer lista de tareas*
2. *Fijar dependencia entre tareas y duración*

Tarea	Precedida por	Duración
A	-	2
B	-	5
C	A	4
D	B	7
E	B	4
F	C-D-E	5
G	E	4

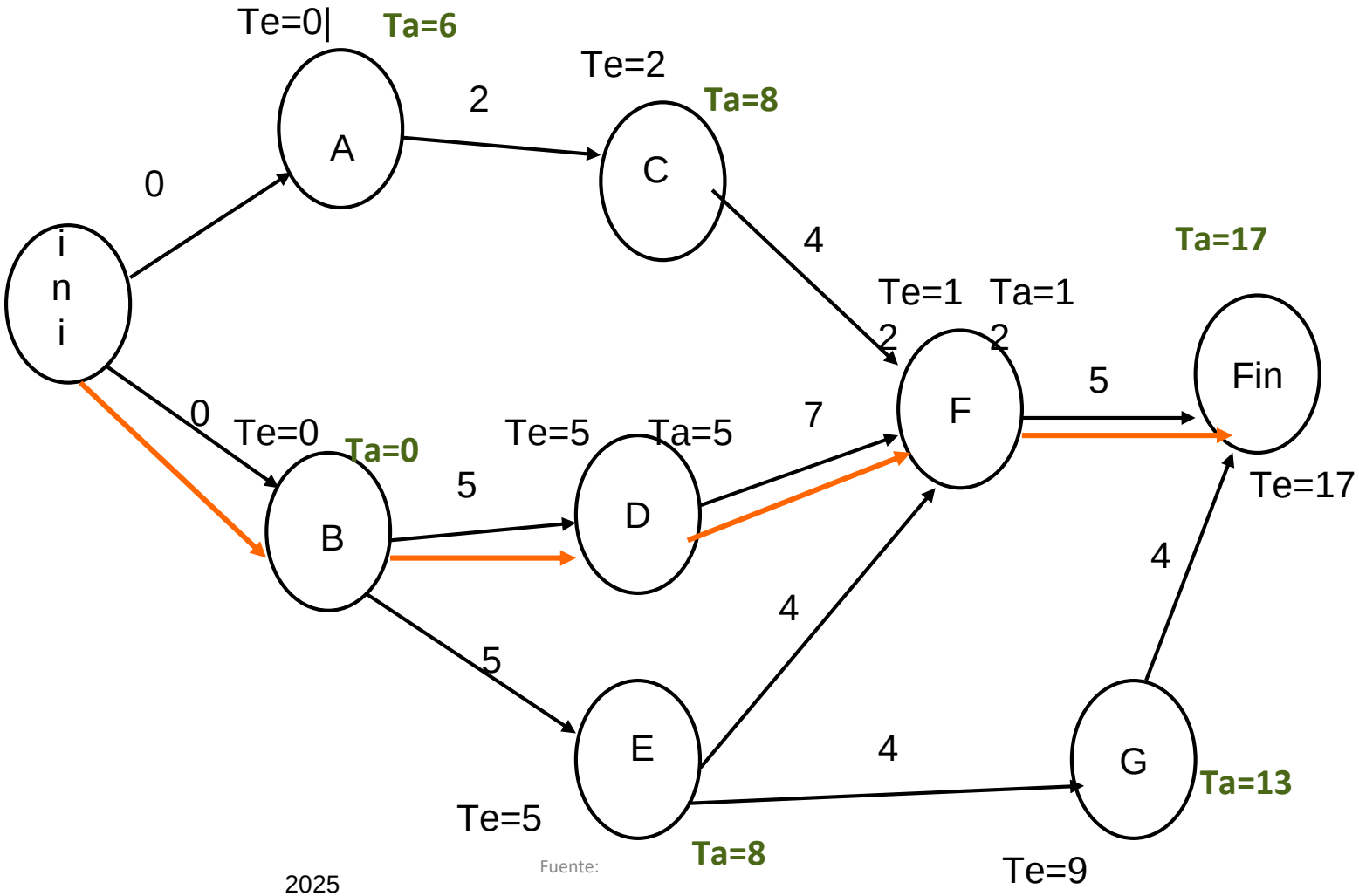
Fuente:

Método de planificación temporal
PERT - CPM



Fuente:

Método de planificación temporal
PERT - CPM



Fuente:

Método de planificación temporal PERT - CPM

Tarea	Precedida por	Duración
A	-	2
B	-	5
C	A	4
D	B	7

- **Fechas Tempranas**

TeJ = TeI + tlJ

Donde

TeJ = fecha más temprana del nodo destino

TeI = fecha más temprana del nodo origen

tlJ = duración de la tarea desde el nodo I hasta el nodo J

Si hay más de un camino ... Max (TeJ1, TeJ2..)

Fuente: Munier, Manual de PERT –CPM

Método de planificación temporal
PERT - CPM

❖ Fechas Tardías

TaI = TaJ – tIJ

Donde

TaI = fecha más tardía del nodo origen

TaJ = fecha más tardía del nodo destino

tIJ = duración de la tarea desde el nodo I hasta el nodo J

Si hay más de un camino ... Min (TaJ1, TaJ2..)

Fuente: Munier, Manual de PERT –CPM

Método de planificación temporal
PERT - CPM

❖ Margen Total

Mt = TaJ – TeI - tIJ

Donde

TaJ = fecha tardía del nodo destino

TeI = fecha temprana del nodo origen

tIJ = duración de la tarea desde el nodo I hasta el nodo J

OBSERVAR que el Margen total también puede calcularse como MtJ= TaJ-TeJ

Es decir como la diferencia entre la fecha más tardía y más temprana del mismo nodo

Fuente: Munier, Manual de PERT –CPM

Método de planificación temporal
PERT - CPM

¿Qué ocurre cuando tengo un margen total de por ej. 6 días?

Significa que la tarea puede iniciarse con 6 días de retraso sin que ello afecte a la duración total del proyecto.

Fuente: Munier, Manual de PERT –CPM

Método de planificación temporal
PERT - CPM

- **¿Qué ocurre cuando el margen total es 0?**

  Significa que no hay margen y que esa tarea hay que iniciarla y finalizarla en las fechas más tempranas.

  Puntualmente estas tareas con margen cero serían críticas.

- El camino formado por una sucesión de tareas críticas recibe el nombre de camino crítico.
- El camino crítico puede obtenerse utilizando el cálculo del margen total.

Fuente: Munier, Manual de PERT –CPM

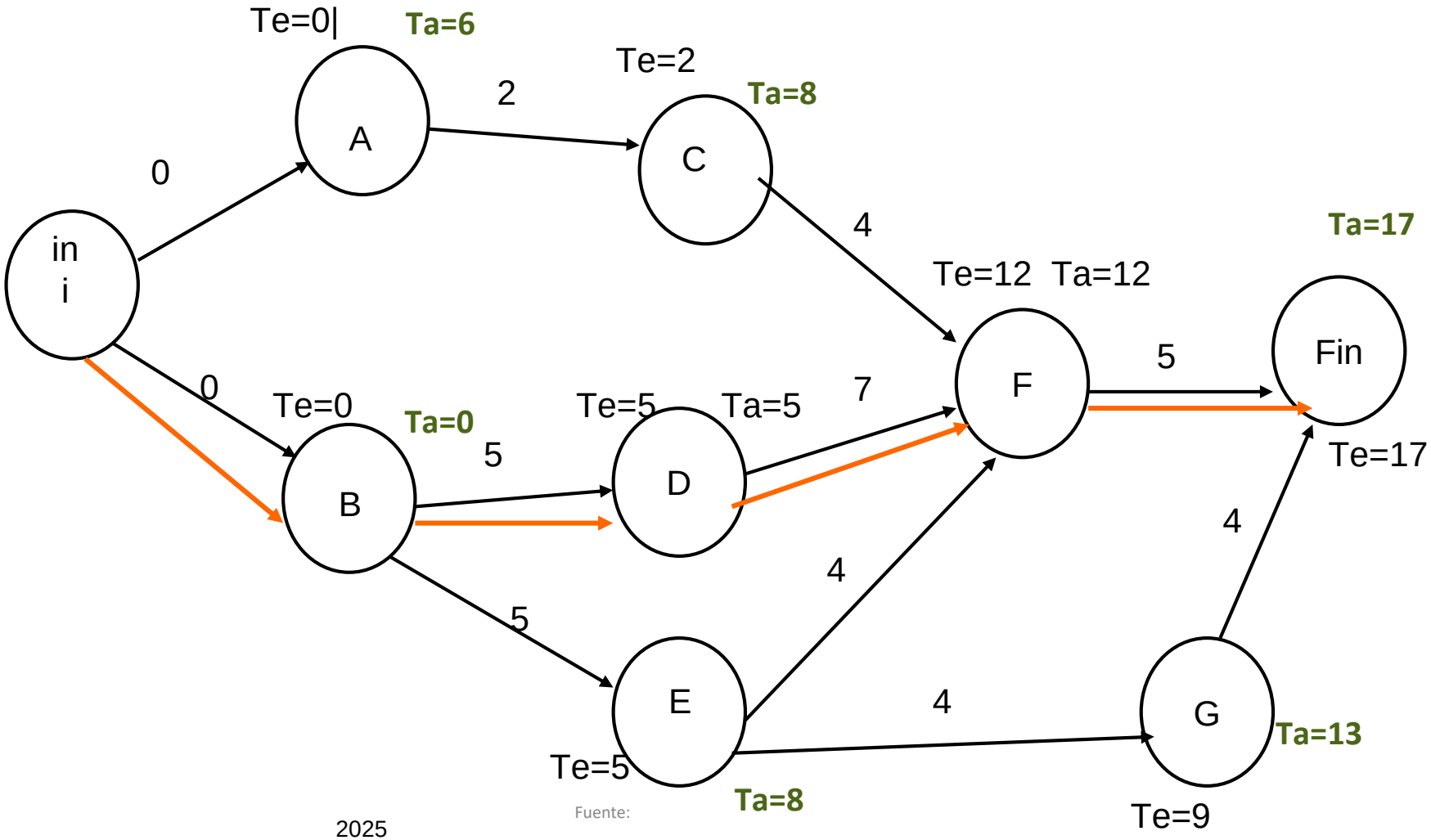
Método de planificación temporal
PERT - CPM

Ejemplo

Tarea	Precedida por	Duración
A	-	2
B	-	5
C	A	4
D	B	7
E	B	4
F	C-D-E	5
G	E	4

Fuente:

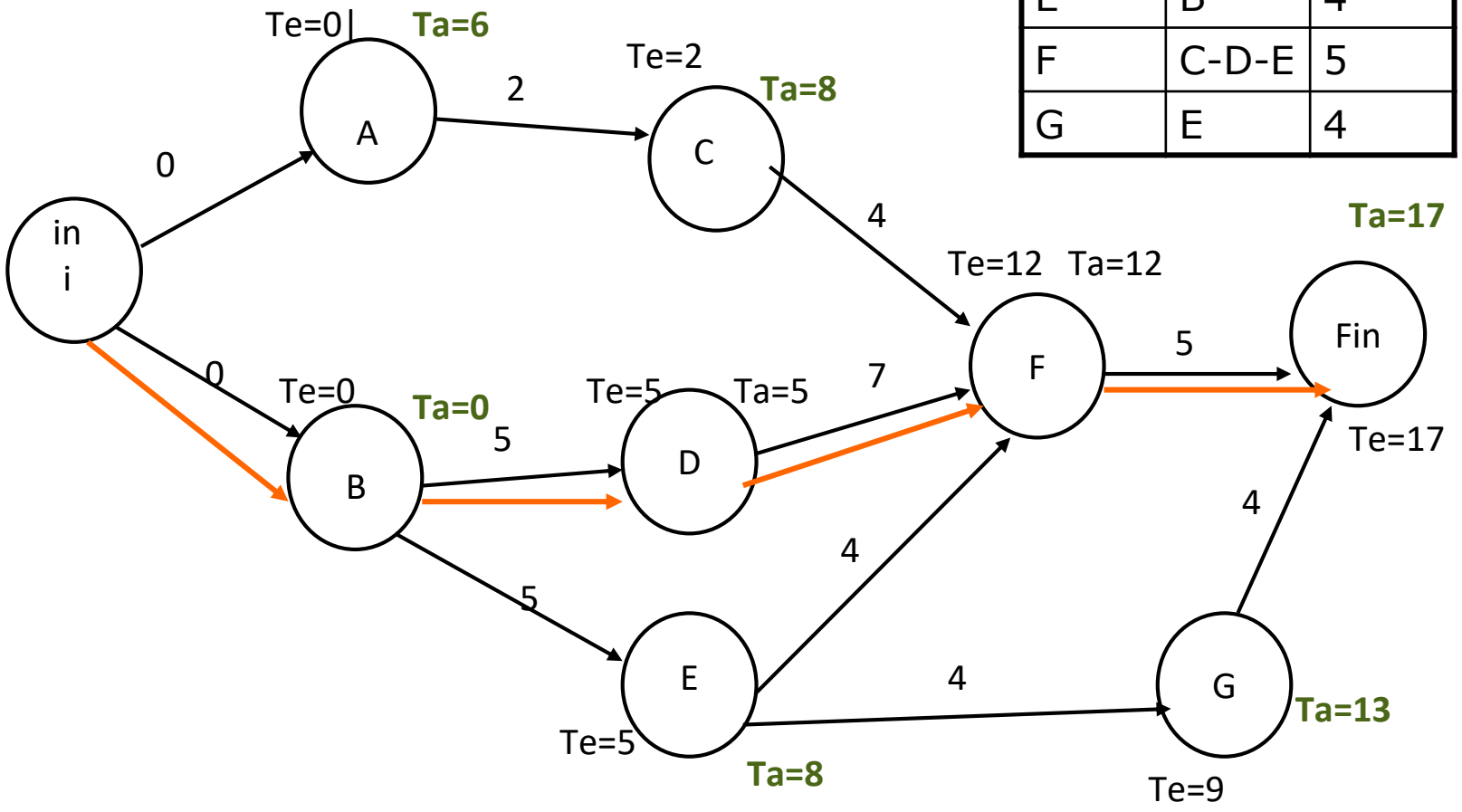
Método de planificación temporal
PERT - CPM



Fuente:

Método de planificación temporal
PERT - CPM

Tarea	Prece d.	Dur.
A	-	2
B	-	5
C	A	4
D	B	7
E	B	4
F	C-D-E	5
G	E	4



Fuente:

Método de planificación temporal
PERT - CPM

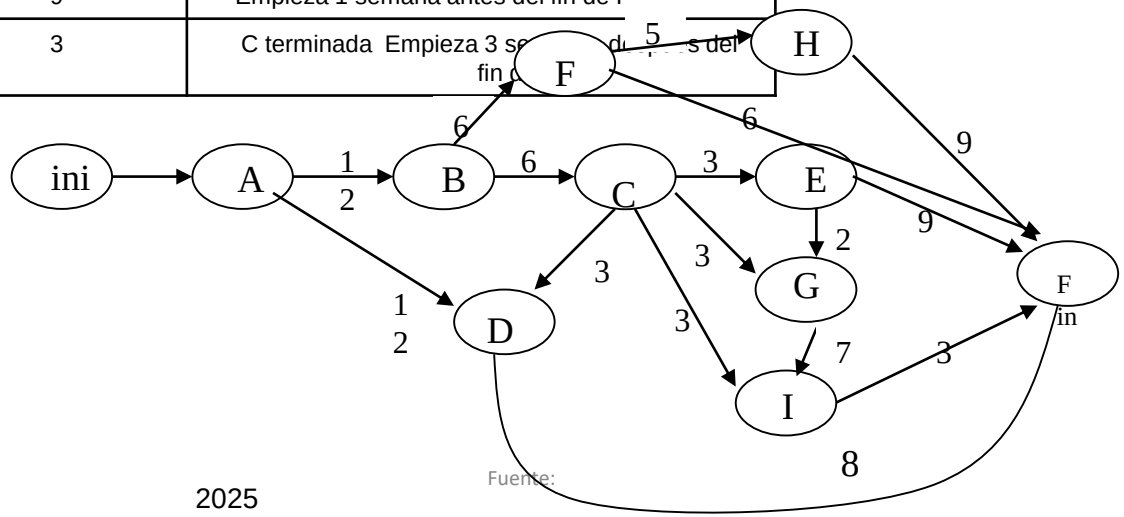
Tarea	Duración	Restricciones
A	12	
B	5	A terminada
C	3	Empieza 1 semana después de terminada B
D	8	A terminada C terminada
E	9	C terminada
F	6	Empieza 6 semanas después del comienzo de B
G	4	C terminada. Empieza 2 semanas después del comienzo de E
H	9	Empieza 1 semana antes del fin de F
I	3	C terminada Empieza 3 semanas después del fin de G

Fuente:

Método de planificación temporal
PERT - CPM

Ejemplo

Tarea	Duración (semanas)	Restricciones
A	12	
B	5	A terminada
C	3	Empieza 1 semana después de terminada B
D	8	A terminada C terminada
E	9	C terminada
F	6	Empieza 6 semanas después del comienzo de B
G	4	C terminada. Empieza 2 semanas después del comienzo de E
H	9	Empieza 1 semana antes del fin de F
I	3	C terminada Empieza 3 se[illegible] d[illegible]s del fin [illegible]

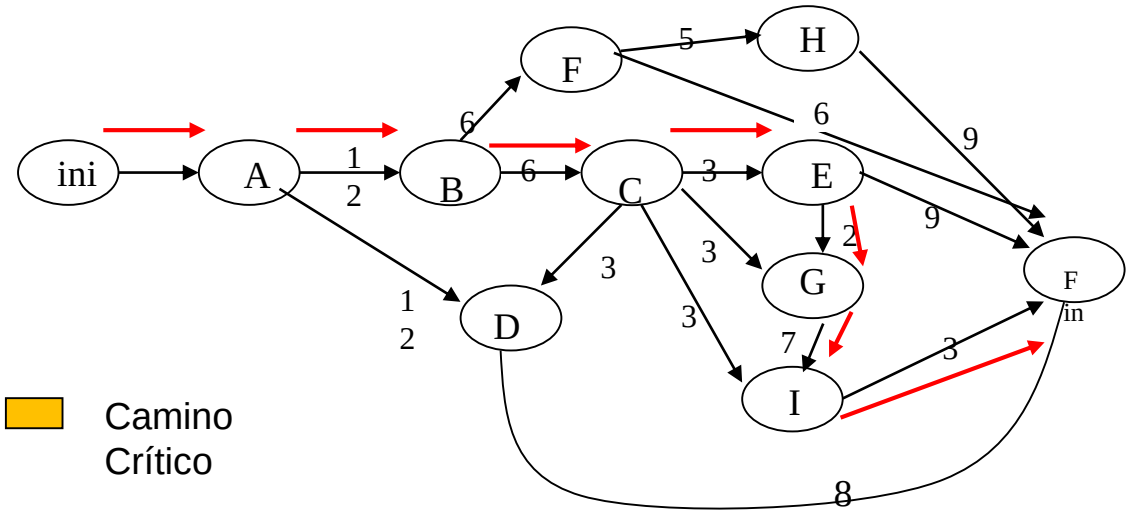


Fuente:

PERT - CPM

Tarea	Duración (semanas)	Restricciones	Te	Ta
A	12		0	0
B	5	A terminada	12	12
C	3	Empieza 1 semana después de terminada B	18	18
D	8	A terminada C terminada	21	25
E	9	C terminada	21	21
F	6	Empieza 6 semanas después del comienzo de B	18	19
G	4	C terminada. Empieza 2 semanas después del comienzo de E	23	23
H	9	Empieza 1 semana antes del fin de F	23	24
I	3	C terminada Empieza 3 semanas después del fin de G	30	30
Fin	0	-	33	33

Ejemplo



Fuente:

PERT-CPM


Datos que se obtienen del PERT - CPM
Camino crítico
El camino crítico determina las tareas esenciales para el proyecto.
Ventana temporal
Indica el tiempo disponible para cada actividad.
Fecha temprana inicio
Muestra cuándo puede comenzar una tarea.
Fecha tardía inicio
Indica el último momento para iniciar sin retrasos.
Final más temprano
Define el momento más pronto en que puede concluir una tarea.
Final más tardío
Establece el último momento para finalizar una tarea.
Margen total
Representa el tiempo total que se puede retrasar sin afectar el proyecto.

Planificación temporal

❖ ¿QUÉ HACER CUANDO UNA TAREA SE SALE DE LA AGENDA?

Revisar el impacto sobre la fecha de entrega

Reasignar recursos

La inclusión de más personas en el desarrollo no siempre genera aumento en la productividad

Reordenar tareas

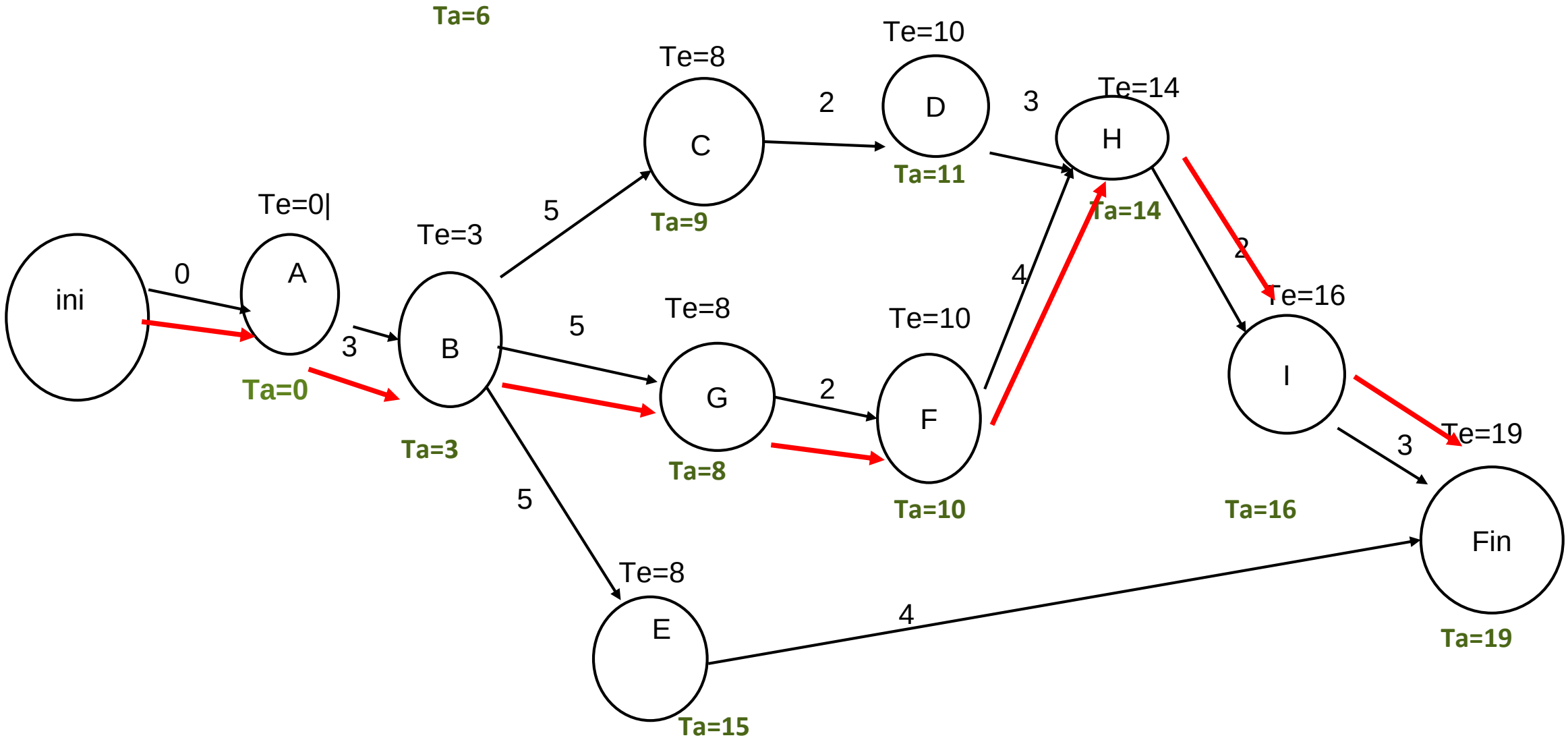
Modificar entrega

Fuente:

Ejemplo

ID	DESCRIPCION	PREDEC	TIEMPO
A	Hacer los cimientos		3
B	Erigir la estructura	A	5
C	Poner las vigas del techo	B	2
D	Revestir el techo	C	3
E	Cableado eléctrico	B	4
F	Paredes exteriores	G	4
G	Colocar ventanas	B	2
H	Paredes interiores	D,F	2
I	Pintura exterior e interior	F,H	3

Método de planificación temporal
PERT - CPM

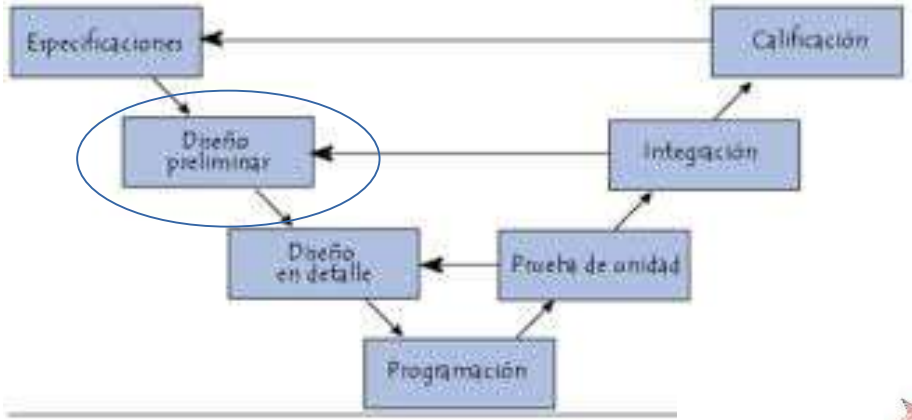


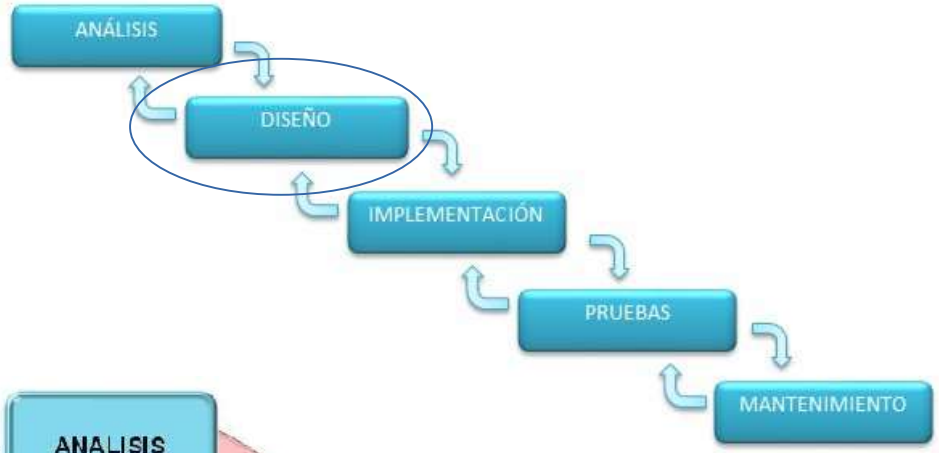
Fuente:

Diseño de Software - Conceptos

Ingeniería de software II - 2025

Etapa de Diseño en los modelos de desarrollo de Software





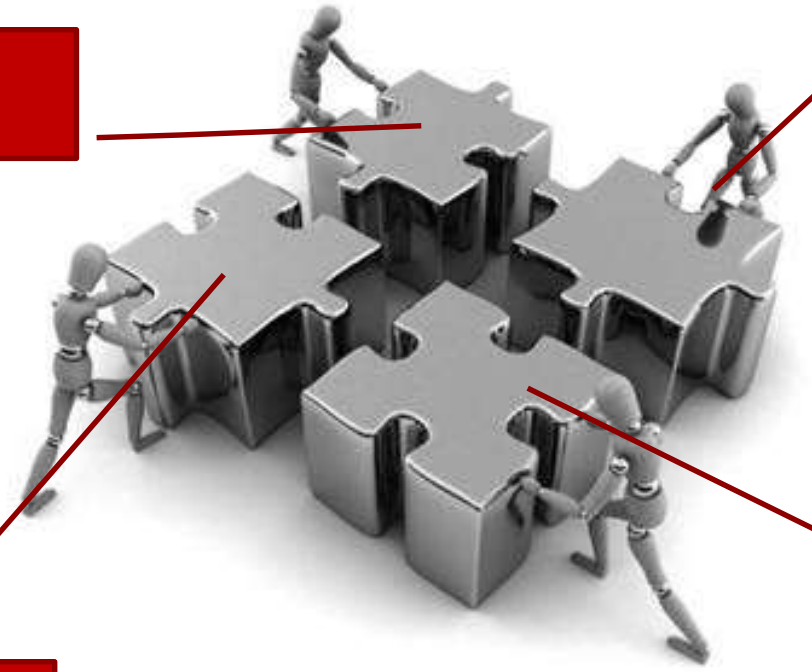
Diseño de Software

¿Qué es?

Áreas

¿Por qué es importante?

Diseño de Software - Tipos



Diseño de datos

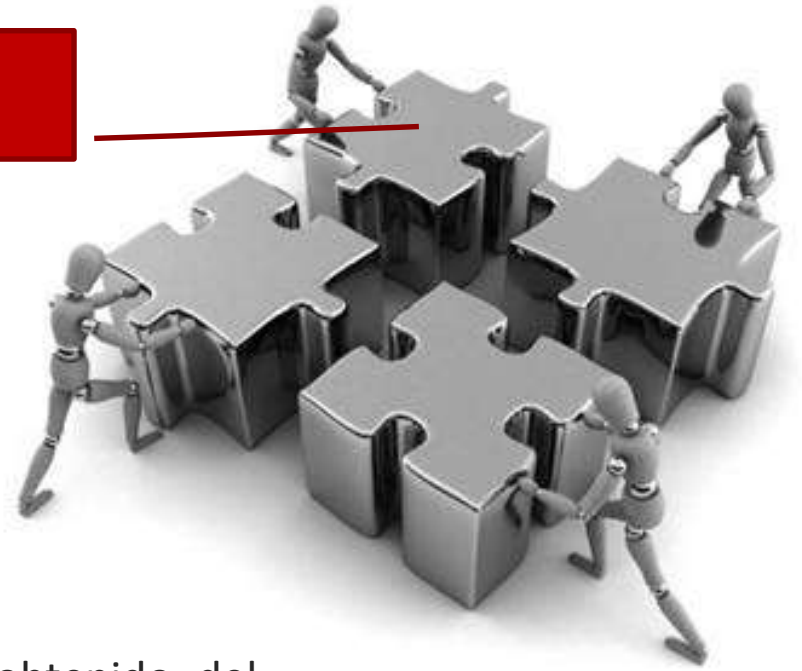
Diseño arquitectónico

Diseño de interface

Diseño a nivel componentes

Diseño de Software - Tipos

Diseño de datos



- Transforma el modelo del dominio, obtenido del análisis, en estructuras de datos, objetos de datos, relaciones , etc.

Diseño de Software - Tipos

Diseño arquitectónico

- Define la relación entre los elementos estructurales del software, los estilos arquitectónicos, patrones de diseño, etc.

Diseño de Software - Tipos

- Transforma los elementos estructurales de la arquitectura de software en una descripción procedimental de los componentes del software.

Diseño a nivel componentes

Diseño de Software - Tipos

- Describe la forma de comunicación dentro del mismo sistema, con otros sistemas, y con las personas.

Diseño de interface

Diseño de Software - Características para su evaluación

- Deberá *implementar* todos los requerimientos explícitos del modelo de requerimientos, e *incorporar* todos los requerimientos implícitos que desea el cliente.
- Deberá ser una *guía* legible y comprensible para aquellos que generan código y para aquellos que dan soporte al software.
- Deberá proporcionar una imagen/visión completa del software. (Completitud)

Criterios técnicos para un buen diseño

1. Deberá presentar una estructura arquitectónica que:

   Se haya creado mediante patrones de diseño reconocibles, que esté formado por componentes con buen diseño y se implemente en forma evolutiva.

1. Deberá ser modular.
2. Deberá contener distintas representaciones.
3. Deberá conducir a estructuras de datos adecuadas y que procedan de patrones de datos reconocibles.
4. Deberá conducir a componentes que presenten características funcionales independientes.
5. Deberá conducir a interfaces que reduzcan la complejidad de las conexiones entre los módulos y con el entorno externo
6. Deberá derivarse mediante un método repetitivo y controlado por la información obtenida durante el análisis de los requisitos del software.
7. Deberá representarse por medio de una notación que comunique de manera eficaz su significado.

Evolución del diseño de software

Es un proceso continuo que abarca mas de seis décadas
Desde Programas modulares y refinamiento de estructuras de software, a enfoques orientado a objetos, orientado a modelos o a pruebas.
Todos tienen características comunes:

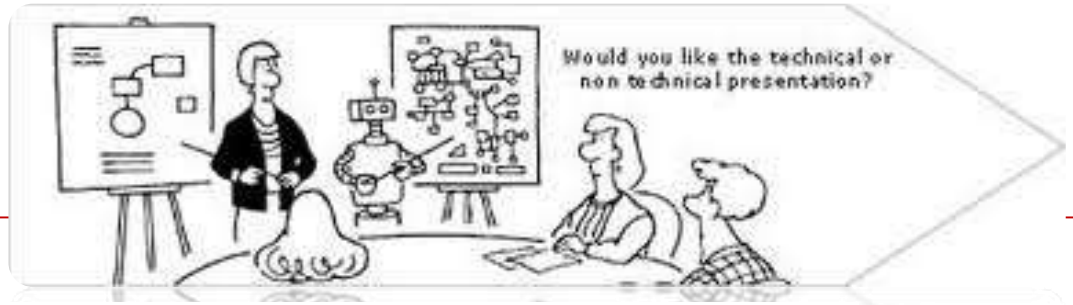
1. un mecanismo para traducir el modelo de requerimientos en una representación de diseño
2. una notación para representar los componentes funcionales y sus interfaces
3. heurísticas para refinamiento
4. lineamientos para evaluación de calidad

Sin importar el tipo de diseño, es necesario aplicar un conjunto de conceptos básicos.

Conceptos de Diseño - Importancia

- ¿Qué criterios se usan para dividir el software en sus componentes individuales?
- ¿Cómo se extraen los detalles de la función o la estructura de datos de la representación conceptual del software?
- ¿Cuáles son los criterios uniformes que definen la calidad técnica de un diseño de software?

Conceptos de Diseño



Conceptos de Diseño

- Abstracción

  Permite concentrarse en un problema a un nivel de *generalización* sin tener en cuenta los detalles de bajo nivel

  Tipos :

Procedimental	*Secuencia* “nombrada” de instrucciones que tienen una funcionalidad específica
De datos	*Colección “nombrada” de datos que definen un objeto real*

Conceptos de Diseño

- Arquitectura del software

  Es la estructura general del software y las formas en que la estructura proporciona una integridad conceptual para un sistema.

  Más adelante se estudiarán las arquitecturas con más detalle

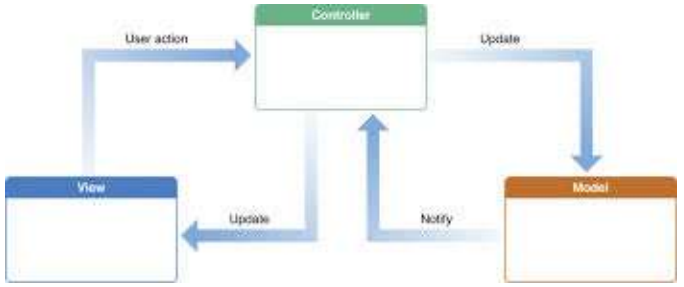
Conceptos de Diseño

- Patrones

  Describen una estructura de diseño que resuelve un problema particular dentro de un contexto específico.

  Deben proporcionar una descripción que permita determinar si

  - ✓es aplicable al trabajo
  - ✓se puede reutilizar
  - ✓puede servir como guía para desarrollar un patrón similar pero diferente en cuanto a la funcionalidad o estructura.



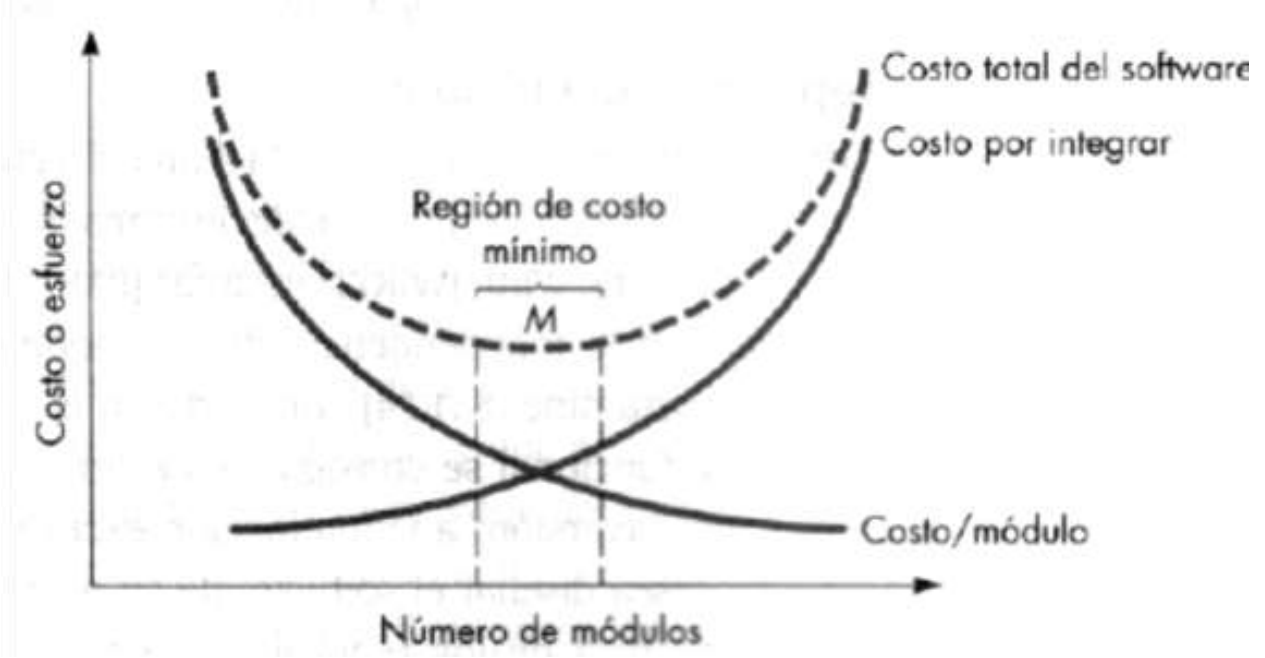
Conceptos de Diseño

- Modularidad

  El software se divide en componentes nombrados y abordados por separado, llamados frecuentemente módulos, que se integran para satisfacer los requisitos del problema.

Conceptos de Diseño

- ¿Cuántos módulos tiene que tener un programa?



Conceptos de Diseño

- Ocultamiento de información

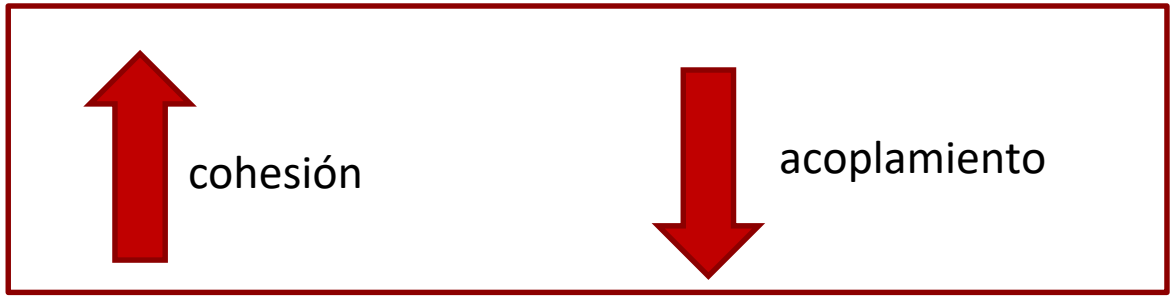
  La información que está dentro un módulo es inaccesible a otros que no la necesiten.

Conceptos de Diseño

Independencia Funcional = *Modularidad + Abstracción + Ocultamiento de Información.*

la cohesión y el acoplamiento entre los módulos

se busca..



Independencia Funcional

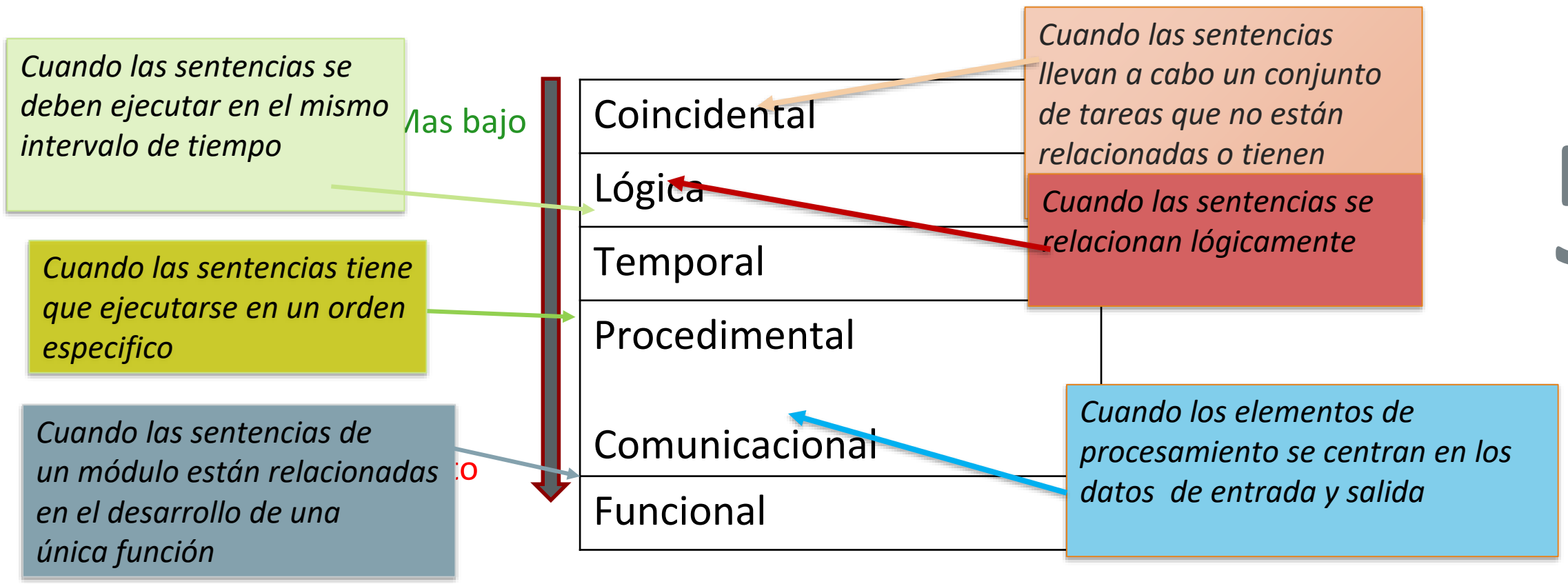
Cohesión (Coherente)

Medida de fuerza o relación funcional existente entre las sentencias o grupos de sentencias de un mismo módulo.

Alta cohesión

Baja cohesión

Independencia Funcional - Tipos de Cohesión



Independencia Funcional

Acoplamiento

Es la medida de interconexión entre los módulos

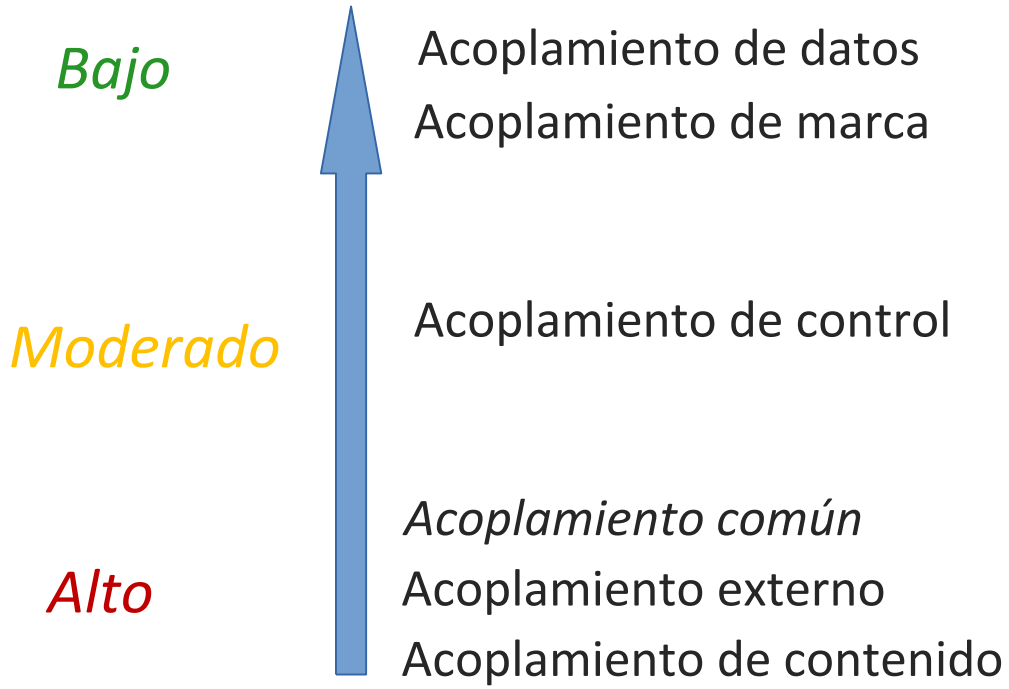
Punto donde se realiza la entrada o referencia y los datos que pasan a través de la interfaz.

Bajo Acoplamiento

Alto Acomplamiento

Independencia Funcional

Niveles de Acoplamiento



Independencia Funcional

Conceptos de Diseño

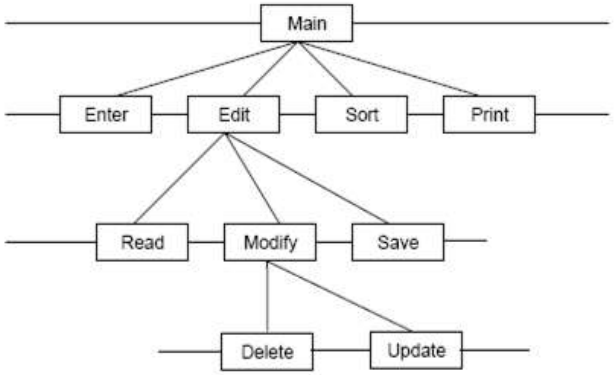
- Refinamiento

  Se refina de manera sucesiva.

La abstracción y el refinamiento son conceptos *complementarios*.

La abstracción permite especificar procedimientos y datos sir considerar detalles de grado menor.

El refinamiento ayuda a revelar los detalles de grado menor mientras se realiza el diseño.



Conceptos de Diseño

- Refabricación o rediseño (Refactoring)

  Técnica de reorganización que simplifica el diseño de un componente sin cambiar su función o comportamiento.

Principios del modelado de diseño

1. El diseño debe poder rastrearse hasta el modelo de requerimientos
2. Considerar siempre la arquitectura del sistema que se va a crear
3. El diseño de los datos es tan importante como el diseño de funciones
4. Las interfaces deben diseñarse con cuidado
5. El diseño de interfaz debe ajustarse a las necesidades del usuario, haciendo énfasis en la facilidad de uso
6. El diseño a nivel de componentes debe ser funcionalmente independiente
7. Los componentes deben tener un bajo acoplamiento
8. Las representaciones de diseño debe entenderse fácilmente
9. El diseño debe desarrollarse de manera iterativa
10. La creación de un modelo de diseño no impide metodología ágil.