

GESTIÓN DE PROYECTOS

Módulo 2: Aspectos básicos de la Gestión de Proyectos

2.3. Gestión del tiempo

Departament d'Organització d'Empreses

Facultat d'Informàtica de Barcelona (FIB)



Gestión del tiempo

1. Introducción
2. Procesos: planificación y organización
3. Herramientas de programación
4. Procesos: actuar y controlar

1. Introducció

¿Qué es?

- **Definición:** Planificar, programar y ejecutar las actividades de un proyecto, de manera que el tiempo total, entre su inicio y finalización, sea el esperado
- Su **gestión** → Procesos que implica...

- Planificación y Organización



- Actuación y Control: Seguimiento y Control

- **Planificar & Programar**

- Planificar: Nivel estratégico → Qué/Cómo/Cuándo respecto al proyecto
- Programación → Técnicas que transforman el Plan de proyecto en un calendario real, que tenga en cuenta recursos, costes, ...

2. Procesos: planificación y organización

Definición de las actividades

- **Matriz de Procesos** (ciclo de vida/tipos de proceso; ver módulo 2.2)
- Se requiere un **proceso de revisión**.
 - **Subdividir procesos** en actividades más concretas, si procede.
 - **Agrupar pequeñas tareas** en actividades más controlables.
 - Ratio **Actividad/Control**
 - ✓ Ajustar el número de actividades en pro de un buen seguimiento
 - ✓ Evitar formalizar tareas de muy corta duración → Mucho esfuerzo de control



2. Procesos: planificación y organización

Ordenación o secuenciación de actividades

- Establecer relaciones de **precedencia** entre actividades: **Final - Inicio**
 - La actividad A debe ser ejecutada antes de las actividades B y C
- Establecer relaciones de **dependencia** entre actividades
 - La actividad D no puede iniciarse hasta que finalicen las actividades B y C
 - Tres tipos de dependencias
 - ✓ **Final – Inicio** (*Precedencia*)
 - ✓ **Inicio – Inicio** (*Dependencia relativa a la ejecución paralela de actividades*)
 - ✓ **Final – Final** (*Dependencia relativa a la ejecución paralela de actividades*)



En el entregable de vuestro proyecto conviene explicar cuáles son la dependencias temporales entre la actividades identificadas (es suficiente trabajar sólo con relaciones de precedencia [Final – Inicio])

2. Procesos: planificación y organización

Estimación de tiempos (I)

- En **base a...**
 - la experiencia
 - información histórica
 - simulaciones
- **Modelo dinámico de estimación** en base a la experiencia e información histórica
 - Clasificar las tareas en 3 niveles de riesgo: **Alto / Medio / Bajo**
 - Calcular la desviación de cada actividad
 - Calcular la desviación conjunta de cada nivel de riesgo
 - Ratios orientativos desarrollo de Software:
 - ✓ Alto: 250%
 - ✓ Medio: 150%
 - ✓ Bajo: 110%



2. Procesos: planificación y organización

Estimación de tiempos (II)

- Distribución del esfuerzo en un proyecto de “Ingeniería de Software”

Project Scheduling and Tracking. Why Are Projects Late?

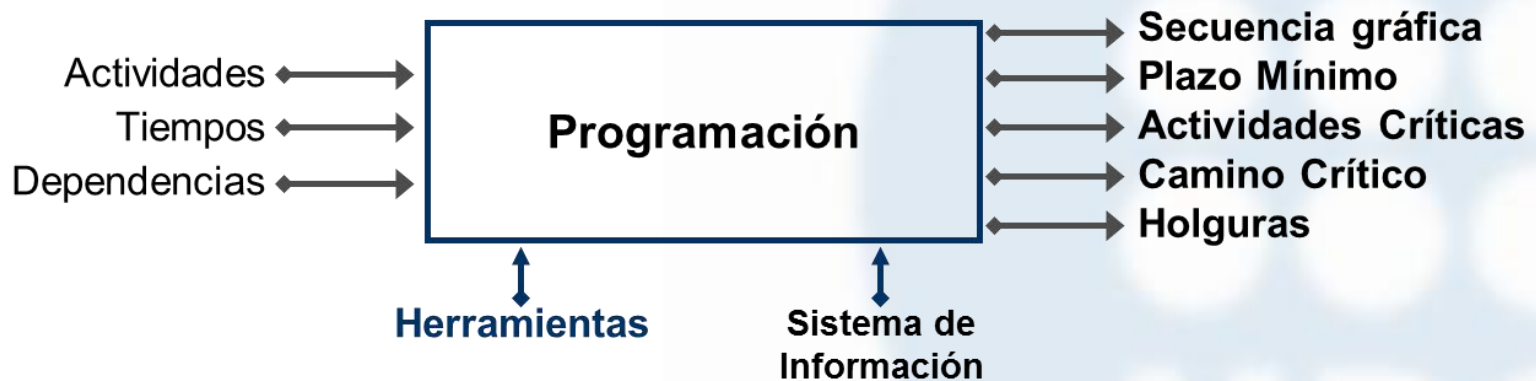
(<http://people.cs.aau.dk/~ivan/SOE2000/SOE08.pdf>)

- **40% - 50%:** Actividades de “Front End”
 - ✓ *Comunicación con los clientes*
 - ✓ *Análisis y diseño*
 - ✓ *Revisión y Modificación*
- **15% - 20%:** Construcción
 - ✓ *Código*
- **30% - 40%:** Pruebas e Instalación
 - ✓ *Pruebas unitarias y de instalación*
 - ✓ *Caja blanca, caja negra*

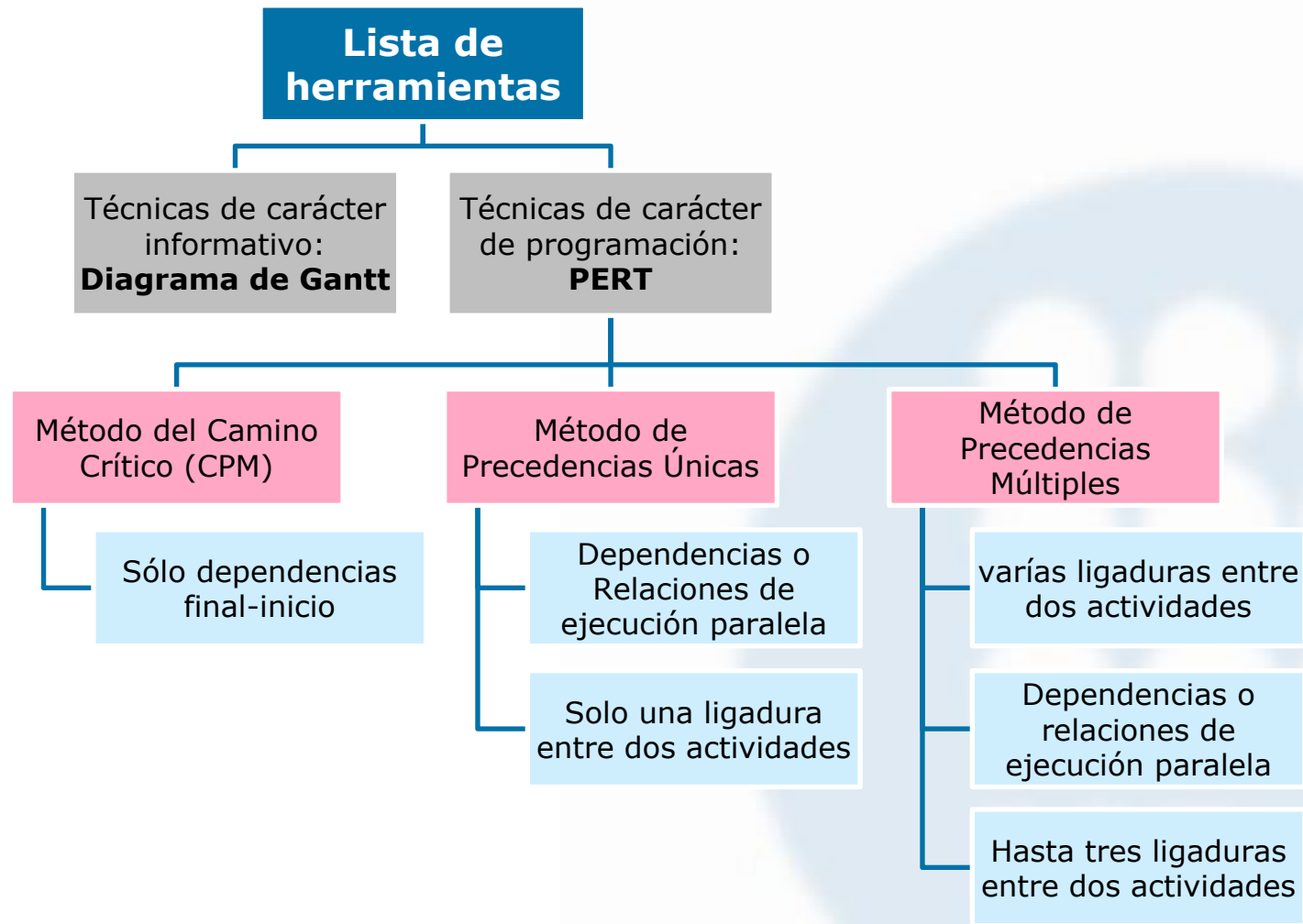
2. Procesos: planificación y organización

La programación de tiempos

- La programación es una herramienta utilizada en el proceso de planificación.
- Se trata de generar un modelo para el conjunto del proyecto, o para una de sus fases, que permita obtener los siguientes resultados.
 - Construcción iterativa
 - Facilitar el control y los cambios por imprevistos



3. Herramientas de programación



3. Herramientas de programación

Diagrama de Gantt > Ejemplos

- **Ejemplo 1:** Se encarga a un albañil abrir un Portón en la valla de nuestra casa, para aparcar el coche en el jardín
 - Las actividades aparecen en el Diagrama
 - Las restricciones son:
 - ✓ *Nada puede hacerse hasta que no se disponga de la licencia.*
 - ✓ *No es posible colocar el portón sin antes derribar el fragmento de valla*
 - ✓ *No es posible reconstruir los extremos de la valla sin haber colocado el portón*
 - ✓ *El contenedor debe estar disponible el día que se derribe la valla*
 - El Proyecto dura 17 días contando los no laborables.
 - Asociar un coste a cada actividad es una buena práctica

3. Herramientas de programación

Diagrama de Gantt > Ejemplos

▪ Solución Ejemplo 1

Nombre de tarea	Duración	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19
Solicitar Licencia	6 días																			
Pedir Contenedor	0 días																			
Derribar Valla	1 día																			
Colocar Porton	2 días																			
Reconstruir Valla	1 día																			
Rebajar realce acera	1 día																			
Rodaduras y muros	2 días																			
Retirnar Contenedor	0 días																			

3. Herramientas de programación

Diagrama de Gantt > Ejemplos

- **Ejemplo 2:** Rediseñar el Diagrama de Gantt anterior hasta obtener un programa en el cual se hayan realizado todas las actividades antes del día 12.
 - Observaciones:
 - ✓ *Puede contratarse a una segunda persona para realizar algunas de las tareas.*
 - ✓ *A partir de la licencia, interesa concluir el trabajo en una semana (cinco días laborables).*
 - ✓ *El contenedor se paga por horas. Mejor retirarlo tan pronto como se haya terminado de derribar la valla y rebajar la acera.*

3. Herramientas de programación

Diagrama de Gantt > Ejemplos

▪ Solución Ejemplo 2

Nombre de tarea	Duración	D-4	D-3	D-2	D-1	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13
Solicitar Licencia	6 días										Esteban							
Pedir Contenedor	0 días																	
Derribar Valla	1 día																	
Colocar Porton	2 días																	
Reconstruir Valla	1 día																	
Rebajar realce acera	1 día																	
Rodaduras y muros	2 días																	
Retirar Contenedor	0 días																	

▪ Implicaciones:

- 3 días de trabajo a cambio de pagar 3 jornales a Ramón
- Reducción de 9 a 1 días el pago por el alquiler del contenedor
- ¿Cuánto cuesta esta optimización en dinero? → Ratio tiempo / coste

3. Herramientas de programación

Diagrama de Gantt > Conclusiones

- **Utilidad**

- Ayuda a detectar y reducir tiempos de espera y tiempos muertos
- Ayuda a equilibrar la carga entre personas
- Muestra fácilmente como se ha planificado el proyecto
- Limitada respecto a facilitar medidas correctoras, si los plazos no se cumplen

- **Ventajas**

- Muy conocido
- Sencillo
- Bueno para visiones globales

- **Desventajas** (y en general de los métodos de carácter informativo)

- No proporciona información sobre la secuencia más óptima de actividades
- No permite deducir el efecto de un retraso en alguna de las actividades

3. Herramientas de programación

Diagrama de Gantt > **Encaje Gantt con Metodologías Ágiles.**

- **A tener en cuenta**

- Cada vez más TFG usan metodologías Ágiles para la gestión del proyecto, en tanto que introduce la filosofía de trabajo expresada en “Agile Manifest”
<https://agilemanifesto.org/>
- Usar **Agile** es compatible con el uso del **Diagrama Gantt**
 - **Agile** permite tratar con un *alcance* más adaptado y variable
 - **Gantt** permite fijar y gestionar la visión global de la planificación de tareas, así como fijar un inicio y final del proyecto

- **Opciones para encajar el Gantt con los Sprints/Iteraciones Agile**

- **Opción 1:** integrar los Sprints/Iteraciones en la dimensión temporal.
- **Opción 2:** integrar los Sprints/Iteraciones en la propia lista de tareas.

3. Herramientas de programación

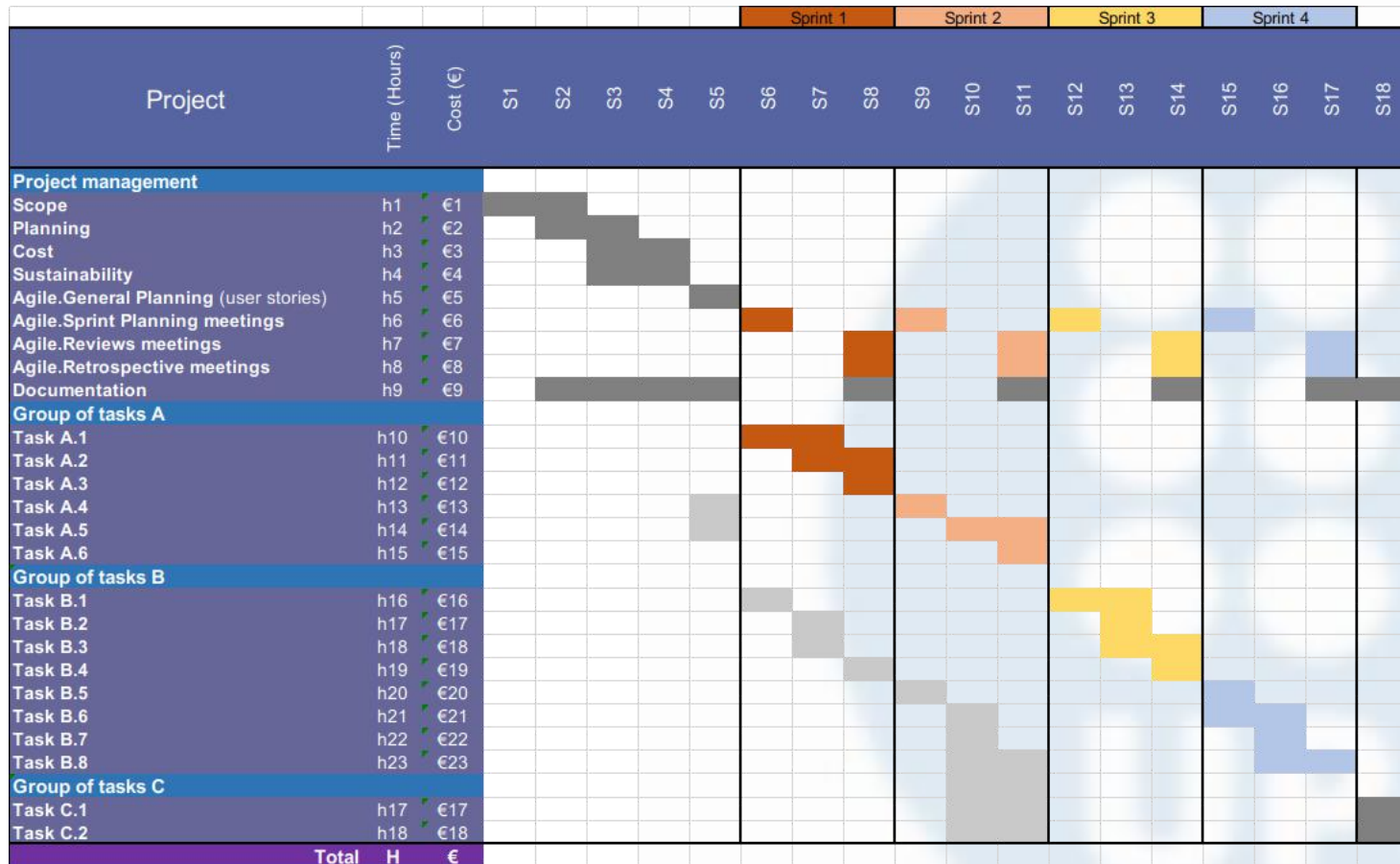
Diagrama de Gantt > **Encaje Gantt con Metodologías Ágiles.**

- **Opción 1:** integrar los Sprints/Iteraciones en la dimensión temporal.
 - **Filas:** Definir la lista de tareas convencionalmente, listando aquellas tareas que son necesarias para implementar la mayoría de las funciones que se estima van a cubrirse.
 - **Columnas:** Definir el vector temporal convencionalmente, introduciendo -por columnas- los intervalos de tiempo que definen la secuencia de Sprints/Iteraciones.
 - Ver ejemplo de estructura en la página siguiente
 - Se utiliza terminología de SCRUM para la parte Agile.

3. Herramientas de programación

Diagrama de Gantt > **Encaje Gantt con Metodologías Ágiles.**

- **Opción 1:** integrar los Sprints/Iteraciones en la dimensión temporal.



3. Herramientas de programación

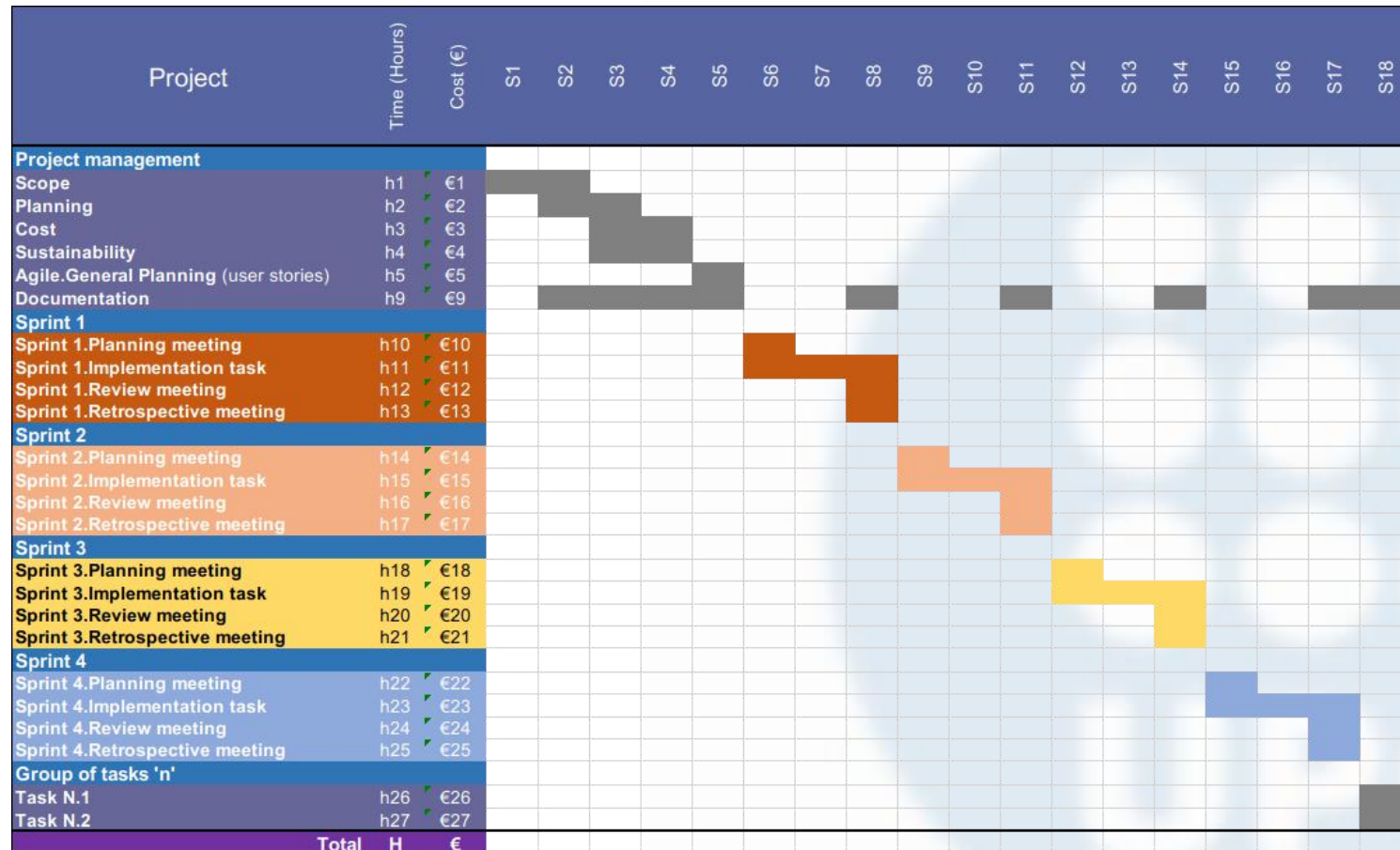
Diagrama de Gantt > **Encaje Gantt con Metodologías Ágiles.**

- **Opción 2** integrar los Sprints/Iteraciones en la propia lista de tareas.
 - **Filas:** Definir la lista de tareas más en abstracto, formalizando los sprints/iteraciones previstos inicialmente.
 - **Columnas:** Definir el vector temporal convencionalmente.
 - Opción válida para alcances inciertos y que sean difíciles de visionar al inicio del proyecto. Es decir, cuando la descripción de las tareas puede variar mucho según lo que se decida implementar en cada sprint.
 - Se aconseja cuando el alcance esté más definido, re-planificar el Gantt siguiendo el esquema de la **opción 1**. En el caso del TFG, esto puede coincidir con el hito de seguimiento o más al final del proyecto.

3. Herramientas de programación

Diagrama de Gantt > **Encaje Gantt con Metodologías Ágiles.**

- **Opción 2** integrar los Sprints/Iteraciones en la propia lista de tareas.



3. Herramientas de programación

Técnicas PERT

- **PERT** = *Project Evaluation Review Technique*
 - Son una evolución del CPM (Critical Path Method)
 - Métodos PERT mejorados
 - ✓ *Método de Precedencias Únicas*
 - ✓ *Método de Precedencias Múltiples*
- Todos ellos representan el proyecto mediante un **grafo**
- **Objetivo:** Programar un proyecto a coste y duración mínimos
 - Definir las dependencias temporales entre actividades
 - Buscar el plazo mínimo para la ejecución
 - Determinar las actividades críticas y las no críticas
 - ✓ *Identificar el camino crítico (secuencia de actividades críticas)*
 - ✓ *Detectar y cuantificar holguras de las actividades no críticas*
 - Herramienta de optimización

3. Herramientas de programación



Técnicas PERT > Metodología

- **Dibujar el grafo**

- Implica conocer los elementos de Construcción
- Implica conocer las reglas de Construcción

- Calcular las **MIC** (fechas Mínimas de Comienzo) para cada Actividad.

- Implica conocer las reglas de Cálculo de las MIC.

- Calcular las **MAC** (fechas Máximas de Comienzo) para cada Actividad

- Implica conocer las reglas de Cálculo de las MAC.

- **Análisis:** Camino Crítico y Holguras

- **Camino Crítico:** Definir y calcular.
- **Holguras:** Definir y calcular.

3. Herramientas de programación

Técnicas PERT > Elementos del grafo

- **Nodos:** Instante que tiene lugar al principio o al final de una tarea
 - Tiene tres datos:
 - ✓ *Identificador: arbitrario*
 - ✓ *MIC: fecha mínima de comienzo. Se calcula una vez dibujado el grafo.*
 - ✓ *MAC: fecha máxima de comienzo. Se calcula una vez dibujado el grafo.*

- Representación:

Id	MIC: n
	MAC: m ($n \leq m$)

- **Flechas:** Actividad que empieza y acaba en los instantes que definen los nodos que la rodean

- Tiene dos datos
 - ✓ *Nombre actividad*
 - ✓ *Tiempo de duración*

- Representación:

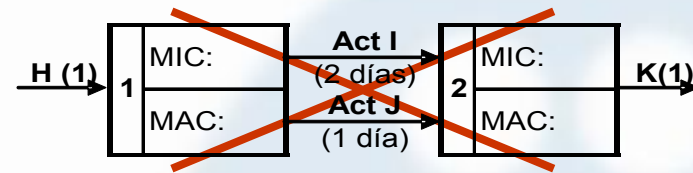
1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>MIC:</td><td></td></tr> <tr><td>MAC:</td><td></td></tr> </table>	MIC:		MAC:		Act A (2 días)	2	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>MIC:</td><td></td></tr> <tr><td>MAC:</td><td></td></tr> </table>	MIC:		MAC:	
MIC:												
MAC:												
MIC:												
MAC:												

3. Herramientas de programación

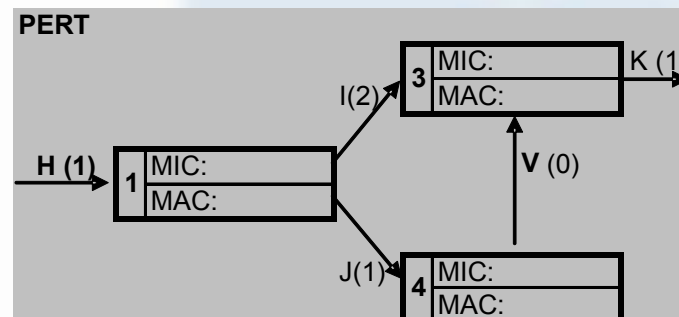
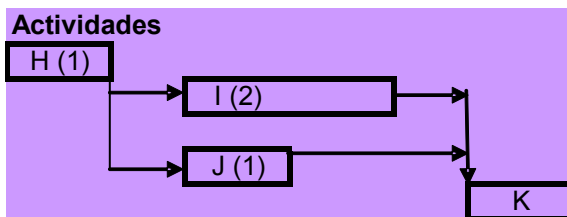
Técnicas PERT > Reglas de construcción del grafo

- Sólo puede haber un nodo inicial y un nodo final.
 - Nodo Inicial: aquel al que no llegue ninguna flecha.
 - Nodo Final: aquel del que no sale ninguna flecha.

- No puede haber nunca dos nodos unidos por más de una flecha.



- Se pueden introducir tareas ficticias, de duración 0, para evitar construcciones ilegales o representar dependencias entre tareas.

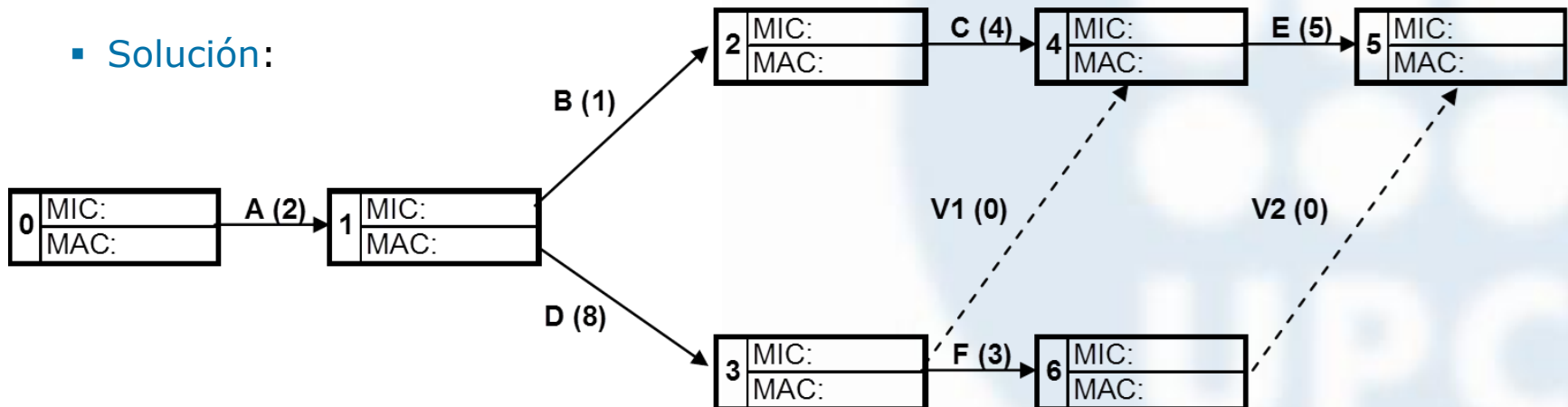


3. Herramientas de programación

Técnicas PERT > Ejemplo 3

- Hay 6 actividades: A, B, C, D, E y F
- Duraciones respectivas: 2, 1, 4, 8, 5 y 3 días
- Dependencias Temporales (tipo final – inicio):
 - A debe acabar antes de iniciar B y D ($A < B, D$)
 - $B < C$
 - $C < E$
 - $D < E, F$

▪ Solución:



3. Herramientas de programación

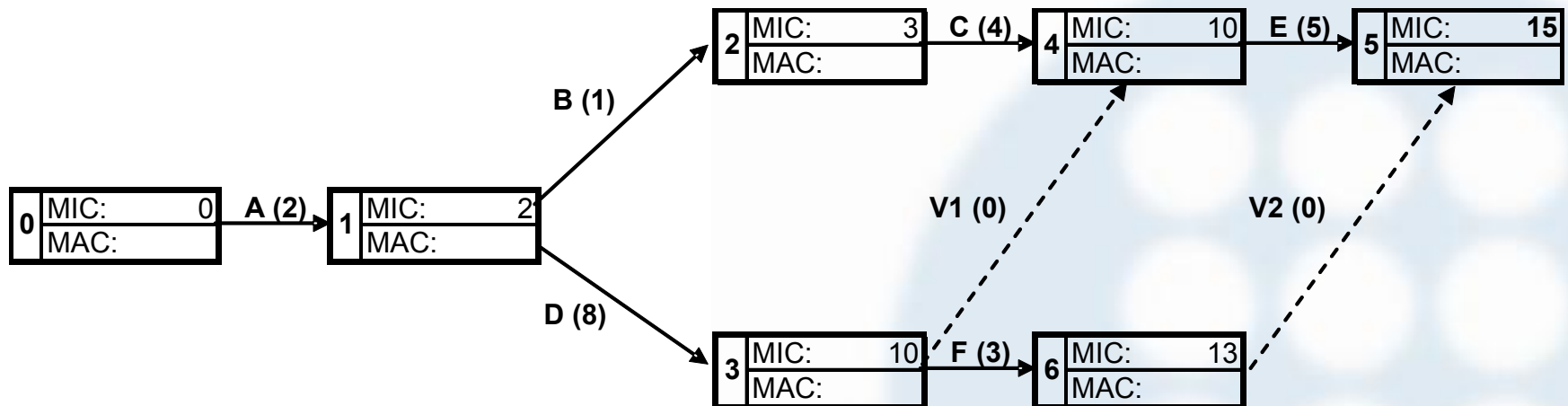
Técnicas PERT > Calcular las MIC

- Calculando la fecha mínima de comienzo de cada actividad para cada suceso, de Izquierda a Derecha, se podrá calcular el MIC del último nodo, lo cual supone calcular el tiempo mínimo de ejecución del proyecto.
- Reglas de cálculo:
 1. Se calcula de izquierda a derecha.
 2. El **suceso inicial** tendrá un **MIC** de **0**, por construcción.
 3. Los MIC de los sucesos contiguos se calculan sumando las duraciones de las actividades entre el suceso inicial y el suceso calculado.
 4. Si en un nodo apuntan varias actividades, su MIC será el **máximo** de los calculados según la regla 3.

3. Herramientas de programación

Técnicas PERT > Ejemplo 4

- En el ejemplo 3, ¿cuál es el tiempo mínimo de ejecución del proyecto?
- Solución:



3. Herramientas de programación

Técnicas PERT > Calcular las MAC

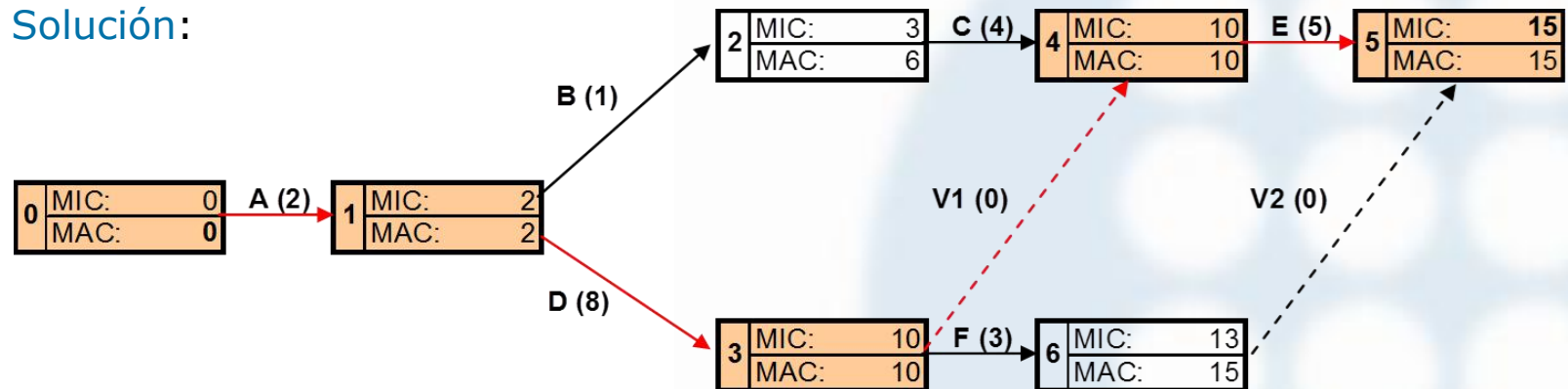
- La fecha máxima de comienzo de actividad calculada en un nodo del grafo, representa la fecha máxima en que debería empezar la siguiente actividad, de manera que no implique un retardo respecto al tiempo mínimo de ejecución calculado.
- Reglas de cálculo:
 1. Se calcula de derecha a izquierda
 2. El **suceso final** tendrá un **MAC igual** al **MIC**, por construcción.
 3. Los MAC de un suceso a la izquierda se calcula como el MAC del suceso de la derecha menos la duración de la actividad que los une.
 4. Si de un nodo se inician varias actividades, su MAC será el **mínimo** de los calculados según la regla 3.

3. Herramientas de programación

Técnicas PERT > Ejemplo 5

- Partiendo del ejemplo 4...
 - Calcular las MAC
 - ¿Cuáles son las actividades clave?
 - ¿Cuál es el camino crítico?

▪ Solución:



- Actividades críticas: A, D, E (no admiten retrasos: MAC=MIC)
- Camino crítico: A-D-E (cualquier retraso en estas actividades, supone no completar el proyecto en 15 días)

3. Herramientas de programación

Técnicas PERT > **Análisis: Camino crítico y holguras**

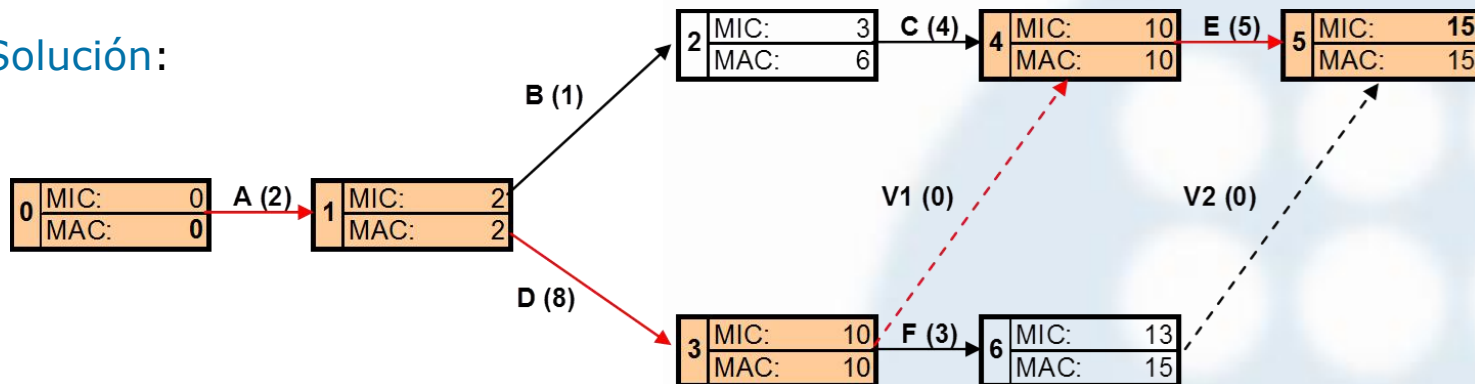
- **Actividades críticas:** Aquellas que no pueden retrasarse.
 - Actividades que, el suceso de inicio y el final tienen igual MIC y MAC.
 - El **camino crítico**: La secuencia de actividades críticas.
 - Un retraso en algunas de las actividades del camino crítico, supone no poder cumplir el tiempo mínimo de ejecución.
- **Actividades no críticas:** Aquellas que pueden retrasarse cierto margen de tiempo, sin afectar al tiempo mínimo de ejecución.
 - Actividades que, el suceso de inicio y el final tienen distinto MIC y MAC.
 - **Holgura**: Margen temporal que puede añadirse a las actividades no críticas, de manera que no retrasen el proyecto en su globalidad.
 - **Cálculo de la holgura** de una actividad (o camino):
 - ✓ $H(\text{Camino}, n_{\text{Inicio}}, n_{\text{Final}}) = \text{MAC}(n_{\text{Final}}) - \text{MIC}(n_{\text{Inicio}}) - \text{Tiempo}(\text{camino})$
 - ✓ $H(\text{ActC}, n_2, n_4) = \text{MAC}(n_4) - \text{MIC}(n_2) - \text{Tiempo}(\text{ActC}) = 10 - 3 - 4 = 3$

3. Herramientas de programación

Técnicas PERT > Ejemplo 6

- Partiendo del ejemplo 3...
 - ¿Cuál es la holgura de la actividad B y de la actividad F?
 - ¿Cuál es la holgura del camino BC?

Solución:



- $H(\text{ActC}, n2, n4) = \text{MAC}(n4) - \text{MIC}(n2) - \text{Tiempo}(\text{ActC}) = 10 - 3 - 4 = 3$
- $H(\text{ActB}, n1, n2) = \text{MAC}(2) - \text{MIC}(1) - \text{Tiempo}(\text{ActB}) = 6 - 2 - 1 = 3$
- $H(\text{ActF}, n3, n6) = \text{MAC}(n6) - \text{MIC}(n3) - \text{Tiempo}(\text{ActF}) = 15 - 10 - 3 = 2$
- $H(\text{CaminoBC}, n1, n4) = \text{MAC}(4) - \text{MIC}(1) - \text{Tiempo}(\text{CaminoBC}) = 10 - 2 - (1 + 4) = 3$
- 3 unidades de tiempo de retardo a repartir entre B y C

3. Herramientas de programación

Técnicas PERT > Conclusiones

▪ Utilidad

- Permite calcular el tiempo mínimo de ejecución de un proyecto.
- Permite detectar Actividades Críticas → Camino Crítico
- Permite calcular y simular situaciones futuras, si se considera el tiempo de una actividad como una variable aleatoria

▪ Ventajas

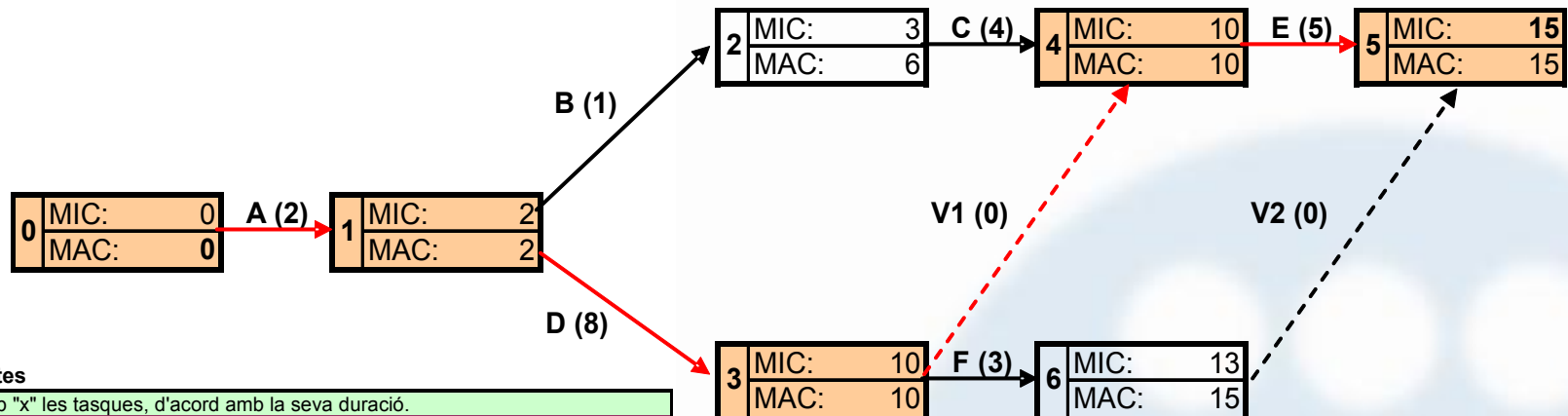
- Muy conocido
- Establece con precisión las dependencias temporales.
- Existen aplicativos y software fáciles de usar que lo calculan y dibujan

▪ Desventajas

- Se requiere ser especialista para entenderlo
- Elementos ficticios: Actividades virtuales
- Un solo tipo de relación entre actividades → Mejoras
 - ✓ *Método de Procedencias únicas: Permite la ejecución paralela de acts.*
 - ✓ *Método de Procedencias múltiples: Más de una ligadura entre dos acts.*

3. Herramientas de programación

Transformación de PERT a GANTT



Notes

Amb "x" les tasques, d'acord amb la seva duració.

Amb "x" (negreta), les tasques corresponents al camí crític.

Amb "o" els marges lliures (de les darreres tasques).

Estimación T C	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A 2	x	x																	A
B 1			x	o	o	o													B
C 4				x	x	x	x	o	o	o									C
D 8			x	x	x	x	x	x	x	x									D
E 5											x	x	x	x	x				E
F 3											x	x	x	o	o				F

3. Herramientas de programación

Para mayor información sobre la construcción de PERTs

Conviene leer el documento:

- Modul 2.3 - Construcción Mejorada de PERTS.pdf

4. Procesos: actuar y controlar

Seguimiento y control

- Comparar los tiempos reales con los programados
- A efectos prácticos
 - Introducir en el modelo los tiempos reales
 - Analizar las consecuencias: holguras (márgenes), camino crítico
 - Anticipar Problemas
- Toma de decisión
 - Decidir cambios



Material

Los materiales de este módulo han sido elaborados gracias a:

- Ferran Sabaté, profesor DOE, UPC



Bibliografia

- Castro, M. et al. (2007). *Gestión de proyectos con Microsoft Project 2007*. Madrid: RA-MA, DL.
- Nokes, S.; Greenwood, A. (2007). *La Guía definitiva de la gestión de proyectos: la vía rápida de todo ejecutivo para cumplir a tiempo y dentro del presupuesto*. Madrid: Prentice Hall Financial Times.
- Rodríguez, J.R.; García, J.; Lamarca, I. (2007). *Gestión de Proyectos Informáticos: métodos, herramientas y casos*. Barcelona: Editorial UOC.

Herramientas online (gratuitas)

- GANTTER (*Chrome web store*)

