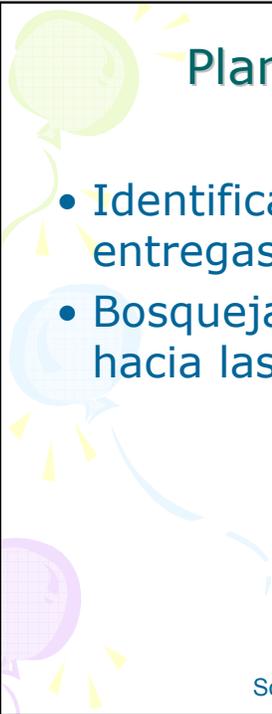


**Planeación  
(de diccionario)**

- **Plan:**
  - Escrito en que sumariamente se precisan los detalles para realizar una obra.
- **Planear:**
  - Trazar o formar el plan de una obra.
  - Hacer planes o proyectos
- **Planeación (Planeamiento):**
  - Acción y efecto de planear

This slide has a decorative background with three lightbulbs in green, blue, and purple. The title "Planeación (de diccionario)" is centered in a bold, dark blue font. The content is organized into a bulleted list with sub-bullets.



## Planeación de proyectos (de software)

- Identificación de actividades, hitos y entregas producidas por un proyecto.
- Bosquejar plan para guiar desarrollo hacia las metas del proyecto

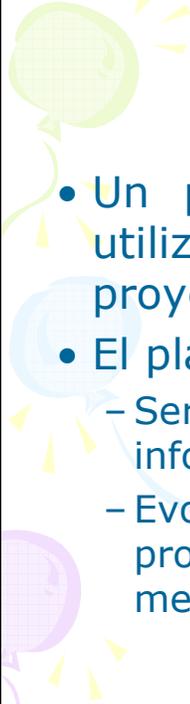
Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



## Planeación (1/4)

- La administración efectiva de un proyecto de software depende de planear completamente el progreso del proyecto.
- Para ello el administrador debe:
  - Anticiparse a los problemas que podrían presentarse.
  - Preparar soluciones tentativas a esos problemas.

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



## Planeación (2/4)

- Un plan, preparado al inicio, será utilizado como conductor para el proyecto.
- El plan inicial deberá:
  - Ser el mejor posible de acuerdo a la información disponible.
  - Evolucionar conforme el proyecto progresa. La información disponible será mejor.

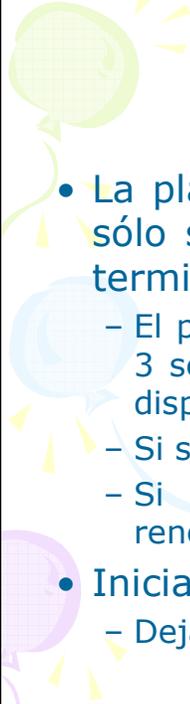
Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



## Planeación (3/4)

1. El proceso de planeación inicia con una valoración de las restricciones que afectan el proyecto, como serían:
  - fecha de entrega requerida, personal disponible y presupuesto global
2. Luego se aplican estimaciones
  - tamaño, esfuerzo, tiempo
3. Se decide enfoque de desarrollo
  1. Lineal, iterativo, espiral
4. Se analizan riesgos
5. Se calendariza

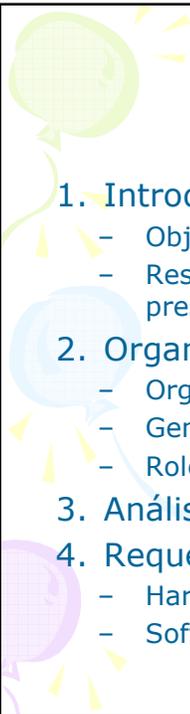
Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



## Planeación (4/4)

- La planeación es un proceso iterativo que sólo se completa cuando el proyecto está terminado.
  - El plan debe revisarse regularmente (cada 2 ó 3 semanas), conforme a la nueva información disponible.
  - Si se requiere, actividades de corrección
  - Si hay problema, revisión técnica y renegociación
- Inicialmente debe ser más bien pesimista
  - Dejar holgura para contingencias

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



## Plan del proyecto (desarrollo 1/2)

1. Introducción:
  - Objetivos del proyecto
  - Restricciones que afecten al proyecto: tiempo, presupuesto y personal disponible
2. Organización del proyecto:
  - Organización del equipo de trabajo
  - Gente involucrada
  - Roles en el equipo
3. Análisis de riesgos (ver tema anterior)
4. Requerimientos de recursos:
  - Hardware (incluir en el precio el que debe comprarse)
  - Software para el desarrollo

## Plan del proyecto (desarrollo 2/2)

### 5. División del trabajo:

- Divide el trabajo en actividades
- Identifica hitos y productos a entregar

### 6. Programa del proyecto:

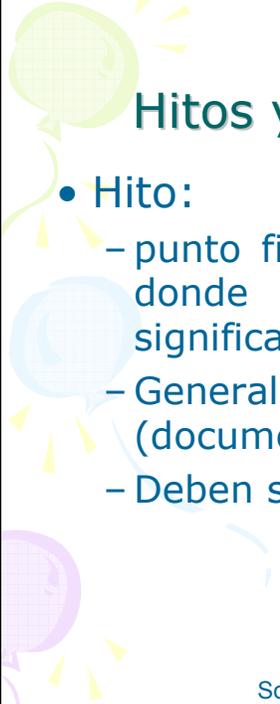
- Describe dependencias entre actividades
- Indica tiempos de logro de un hito
- Asigna responsables

### 7. Mecanismos de supervisión e informe

- Cómo se gestionan informes
- Cuándo se deben entregar
- Describe mecanismos de supervisión

## Tipos de Planes adicionales (no siempre se incluyen)

Plan	Descripción
Plan de Calidad	Describe los procedimientos y estándares de calidad que se utilizarán en un proyecto.
Plan de Validación	Describe el enfoque, los recursos y la programación utilizados para la validación del sistema.
Plan de Administración de la Configuración	Describe los procedimientos de administración de la configuración y las estructuras a utilizarse.
Plan de Mantenimiento	Predice los requerimientos de mantenimiento del sistema, los costos del mantenimiento y el esfuerzo requerido.
Plan de Desarrollo	Describe cómo se desarrollarán las habilidades y experiencia de los miembros del equipo del proyecto.

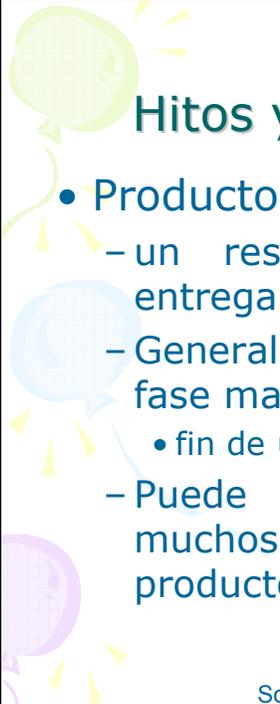


## Hitos y productos a entregar

- Hito:

- punto final de una actividad; momento donde se ha logrado algún aspecto significativo del proyecto.
- Generalmente incluyen salida formal (documento breve).
- Deben ser concretos y verificables.

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley

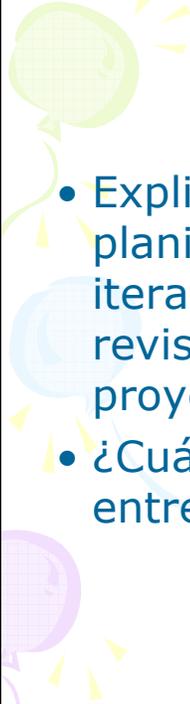


## Hitos y productos a entregar

- Producto a entregar (entregable):

- un resultado del proyecto que se entrega al cliente.
- Generalmente se entrega al final de una fase mayor del proyecto
  - fin de una iteración, al concluir un prototipo
- Puede coincidir con un hito, pero muchos hitos no coinciden con productos.

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



## Ejercicio

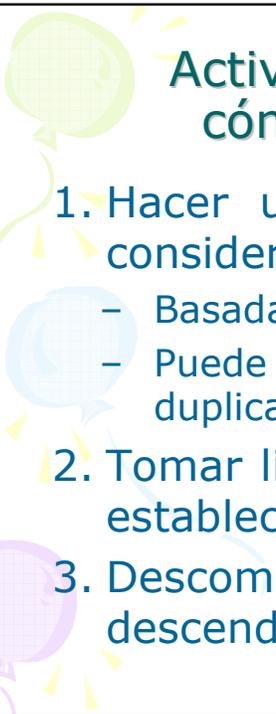
- Explique por qué el proceso de planificación de proyectos es iterativo y por qué un plan se debe revisar continuamente durante el proyecto de Software.
- ¿Cuál es la diferencia fundamental entre un **Hito** y una **Entrega**?

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



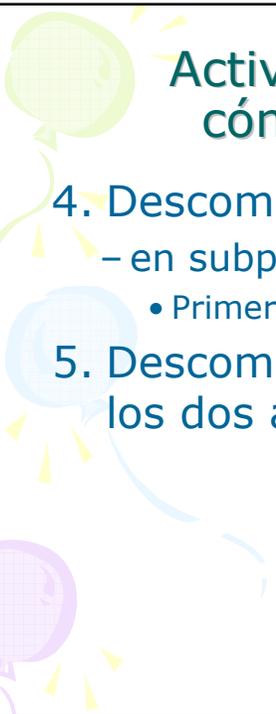
## Elementos de la planeación

- ¿Qué ...
  - Actividades (tareas) se requieren?
  - Hitos se establecen?
  - Productos se entregarán?
- ¿Cuándo ...
  - Comienza y termina cada actividad?
  - Ocurre el hito?
  - Se entrega el producto?



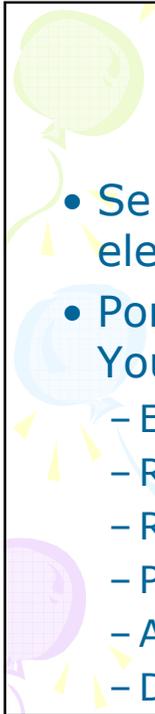
## Actividades a considerar: cómo obtenerlas (1/2)

1. Hacer una lista de tareas que se consideran necesarias;
  - Basada en experiencia y estudios
  - Puede quedar incompleta y con duplicados
2. Tomar lista de un proceso muy bien establecido
3. Descomponer el trabajo de manera descendente



## Actividades a considerar: cómo obtenerlas (2/2)

4. Descomponer el producto
  - en subproductos
    - Primero los que son necesarios para otros
5. Descomponer por entregables (usa los dos anteriores)



## Actividades según proceso

- Se sigue la propuesta del proceso elegido.
- Por ejemplo, para el método de Yourdon podría ser:
  - Entrevistar usuarios
  - Realizar análisis
  - Realizar diseño
  - Preparar pruebas
  - Aplicar pruebas
  - Documentar



## Actividades según proceso

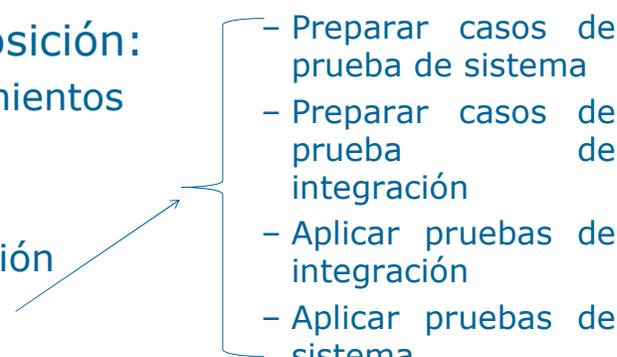
- Análisis se podría descomponer en actividades menores:
  - Modelo ambiental
    - Preparar declaración de propósitos, diagrama de contexto y lista de acontecimientos
  - Modelo de Comportamiento
    - Preparar diagrama de flujo de datos
    - Preparar diagrama entidad-relación
    - Preparar diccionario de datos
    - Preparar diagrama de estados

## Actividades por descomposición descendente del trabajo

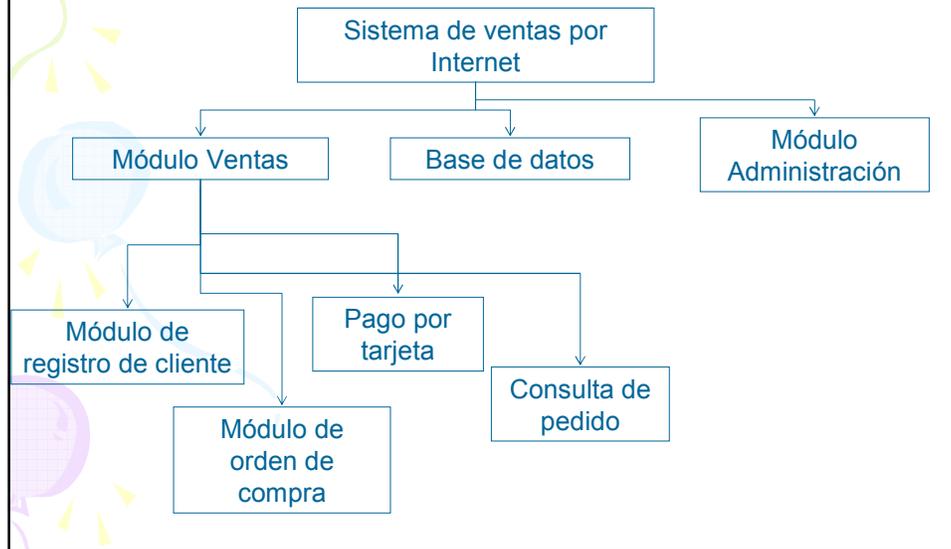
- Puede relacionarse con los otros enfoques, pero parte de lo más general y va detallando poco a poco
- Un ejemplo es la descomposición de Análisis del ejemplo anterior

## Actividades por descomposición descendente del trabajo

Otro ejemplo: (Lineal secuencial). Trabajo: crear un sistema de información ...

- Primera descomposición:
    - Requerimientos
    - Análisis
    - Diseño
    - Codificación
    - Pruebas
  - Segunda descomposición
    - Preparar casos de prueba de sistema
    - Preparar casos de prueba de integración
    - Aplicar pruebas de integración
    - Aplicar pruebas de sistema
- 
- An arrow points from the 'Pruebas' item in the first list to the 'Segunda descomposición' list, indicating that the testing phase is further decomposed into these specific tasks.

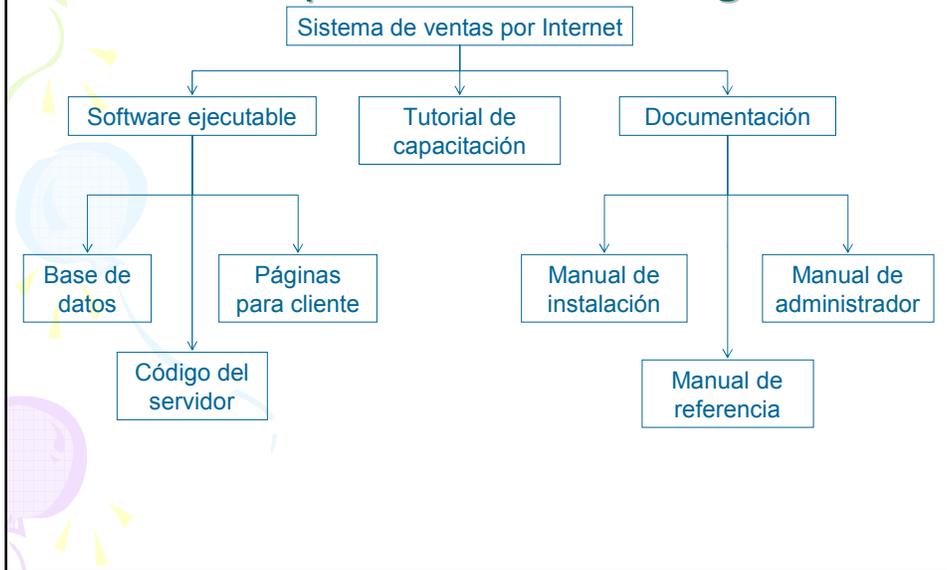
## Descomposición del producto



## Descomposición del producto

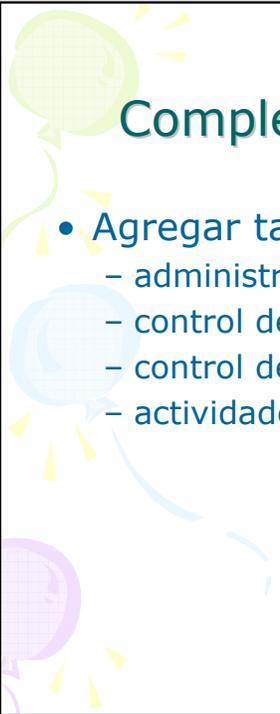
- Actividades derivadas de las partes del producto:
  - Desarrollar módulo de registro de cliente
  - Desarrollar módulo de orden de compra
  - Realizar enlace a sistema bancario
  - Desarrollar módulo de consulta de estado del pedido
- Los productos de los subsistemas sirven a otras partes del sistema:
  - El registro del cliente y la orden , para administración (envío de pedido, factura, etc.)
  - La base de datos para preparar orden de compra
  - El envío (viene de administración) sirve para consulta de estado

## Descomposición de entregables



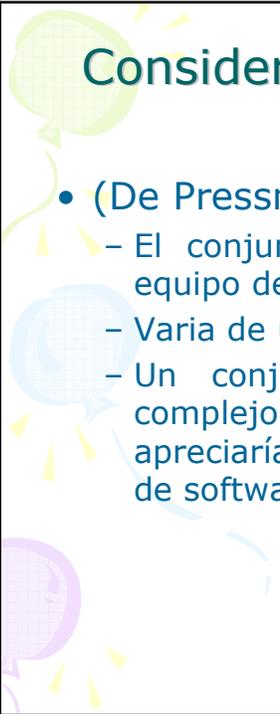
## Descomposición de entregables

- Del árbol anterior se pueden generar actividades:
  - Diseñar la base de datos
  - Desarrollar código del servidor
  - Preparar manual de instalación
  - Etc.



## Completar conjunto de tareas

- Agregar tareas de cobertura
  - administración del proyecto,
  - control de calidad,
  - control de la configuración,
  - actividades para mitigar riesgos



## Consideraciones sobre conjuntos de tareas

- (De Pressman)
  - El conjunto de tareas No debe abrumar al equipo del proyecto
  - Varía de un proyecto a otro
  - Un conjunto de tareas para un sistema complejo y grande probablemente se apreciaría como destructivo para un producto de software pequeño y simple



## Algunos elementos de las actividades

- Cada actividad debe poderse estimar en cuanto a su duración
- Debe poderse estimar los recursos necesarios para la actividad
- Una actividad puede tener precedentes, es decir, actividades que deben terminar antes de comenzar
- Para iniciar una actividad pueden ser necesarios recursos que sólo están disponibles en cierto momento



## Factores que influyen en la elección del conjunto de tareas

1. Tamaño del proyecto
2. Número de usuarios potenciales
3. Duración de la aplicación
4. Estabilidad de los requisitos
5. Facilidad de la comunicación con el usuario o el desarrollador
6. Madurez de la tecnología aplicable
7. Restricciones del desempeño
8. Equipo del proyecto

## Red de tareas para el Proyecto de software

- Una red de tareas o red de actividades, es una representación gráfica del flujo de tareas de un proyecto.
- En ocasiones se utiliza como el mecanismo mediante el cual la secuencia y dependencia entre tareas son la entrada a una herramienta de calendarización del proyecto.
- Las ligas indican precedentes que deben cumplirse antes de iniciar una tarea



Pressman, R., "Ingeniería de Software", 6ª ed., McGraw-Hill

## Características de Red

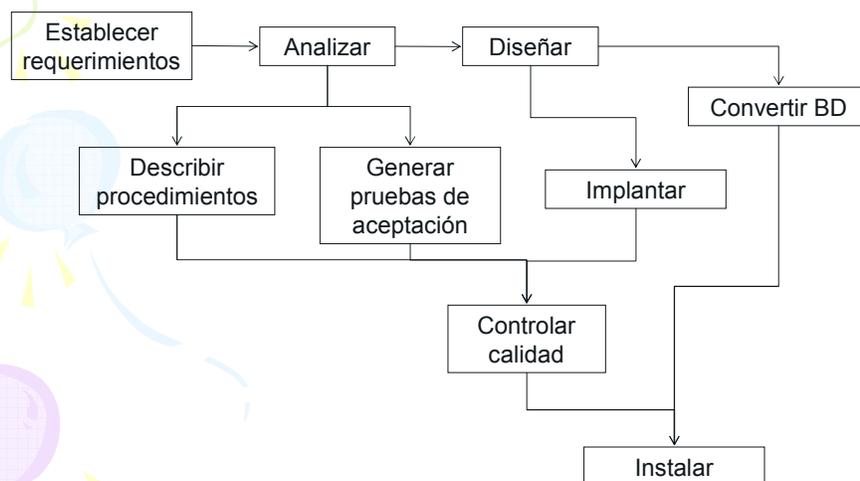
- Sólo una actividad final
- No debe haber ciclos
- No debe haber actividades desconectadas, todas son accesibles desde las tareas iniciales
- Las actividades se inician cuando está lista para ejecutarse al menos una de las tareas iniciales
- Por claridad pueden agregarse actividades nulas (no gastan recursos)
- Una tarea o actividad no inicia si no están terminadas todas las predecesoras

## Red de tareas para el proyecto de software

- Puntos a considerar por el planificador con las actividades concurrentes:
  - Se deben determinar dependencias entre tareas para asegurar el proceso continuo hacia la finalización
  - Estar atento a las tareas que se encuentran en la ruta crítica,
    - poner atención en qué se deben completar en la calendarización.

Pressman, R., "Ingeniería de Software", 6ª ed., McGraw-Hill

## Ejemplo: actividades de Yourdon





## Calendarización y secuenciación (1/2)

- En la secuenciación de tareas se decide en qué orden se realizarán las tareas de un conjunto
  - cumpliendo con sus restricciones
- Una red de tareas puede secuenciarse de diversas maneras según los recursos disponibles
  - Un solo empleado no podrá hacer actividades paralelas y tendrá que hacerlas una a una; si hay más, sí se pueden hacer en paralelas, pero no todas



## Calendarización y secuenciación (2/2)

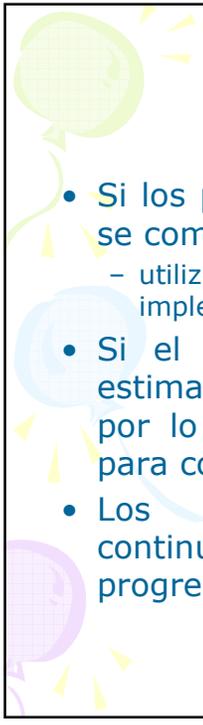
- La calendarización consiste en fijar fechas de inicio y fin a cada actividad
- En proyectos pequeños se hacen al mismo tiempo



## Calendarización (1/5)

- Consiste en estimar tiempos y recursos requeridos para completar las actividades y organizarlas.
- Es muy demandante para los administradores de Software.
  - Si el proyecto es igual a otro anterior se pueden utilizar datos históricos.

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley

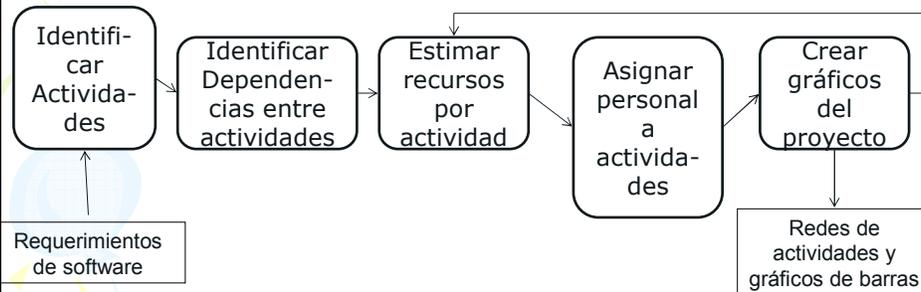


## Calendarización (3/5)

- Si los proyectos son diferentes la calendarización se complica, ya que pueden:
  - utilizar diferentes métodos de diseño y lenguajes de implementación diferentes.
- Si el proyecto es técnicamente complejo, las estimaciones inicialmente resultan optimistas, por lo cual debe considerarse cierto pesimismo para compensar.
- Los calendarios deben actualizarse continuamente a medida que el proyecto vaya progresando.

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley

## Calendarización (4/5)



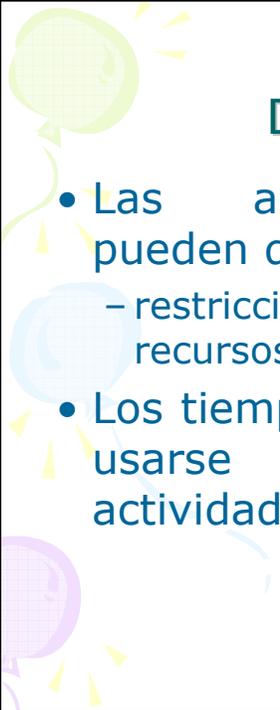
Proceso de Calendarización del proyecto

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley

## Calendarización (5/5)

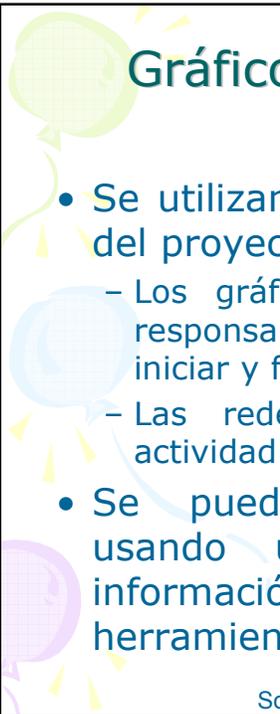
- El calendario del proyecto:
  - se representa como un conjunto de gráficos,
  - muestra la división de trabajo,
  - las dependencias de las actividades y
  - la asignación de personal.
- Herramienta para la gestión de software:
  - Microsoft Project
  - Open Project.

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley



## Datos adicionales

- Las actividades calendarizadas pueden dejar tiempos libres, por:
  - restricciones o por disponibilidad de recursos
- Los tiempos libres (holguras) pueden usarse para atrasar o adelantar actividades



## Gráficos de Barra y Redes de Actividades

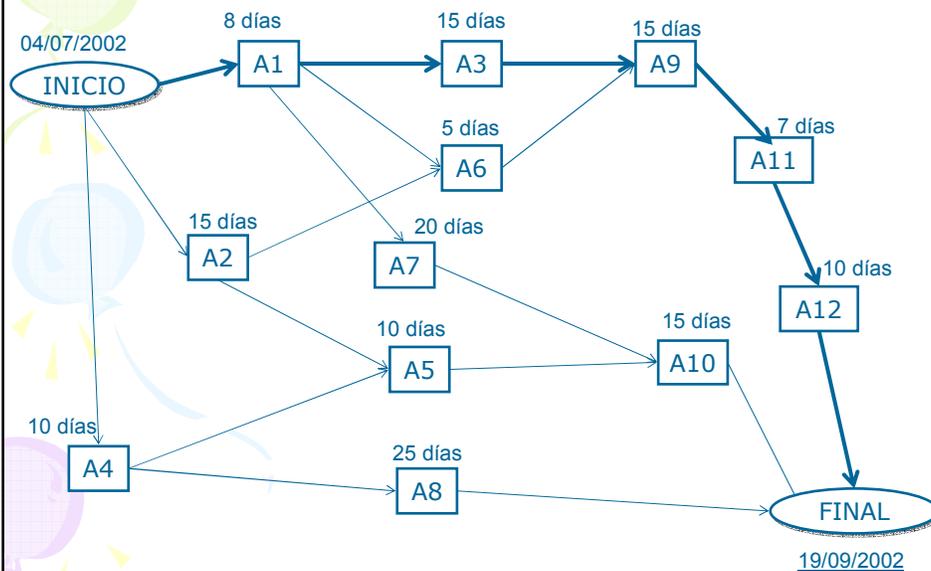
- Se utilizan para ilustrar la calendarización del proyecto.
  - Los gráficos de barras muestran quién es responsable de cada actividad y cuándo debe iniciar y finalizar.
  - Las redes de actividades muestran cada actividad y la dependencia que hay entre ellas.
- Se pueden generar automáticamente usando una base de datos de la información del proyecto utilizando una herramienta de gestión de proyectos.

## Una lista de Duración y Dependencia de Actividades

TAREA	DURACIÓN (DÍAS)	DEPENDENCIAS
A1	8	
A2	15	
A3	15	A1
A4	10	
A5	10	A2, A4
A6	5	A1, A2
A7	20	A1
A8	25	A4
A9	15	A3, A6
A10	15	A5, A7
A11	7	A9
A12	10	A11

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley

## Red de Actividades



Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley

## Ruta crítica

- La ruta crítica es un camino dentro de la red planeada que tiene el menor tiempo posible para terminar; es el camino secuencial más largo.
- Las actividades fuera de la ruta crítica se pueden acomodar para reducir tiempos o usar mejor los recursos.
- En el ejemplo anterior es la ruta marcada más gruesa y es de 55 días.

## Gráfico de Barras

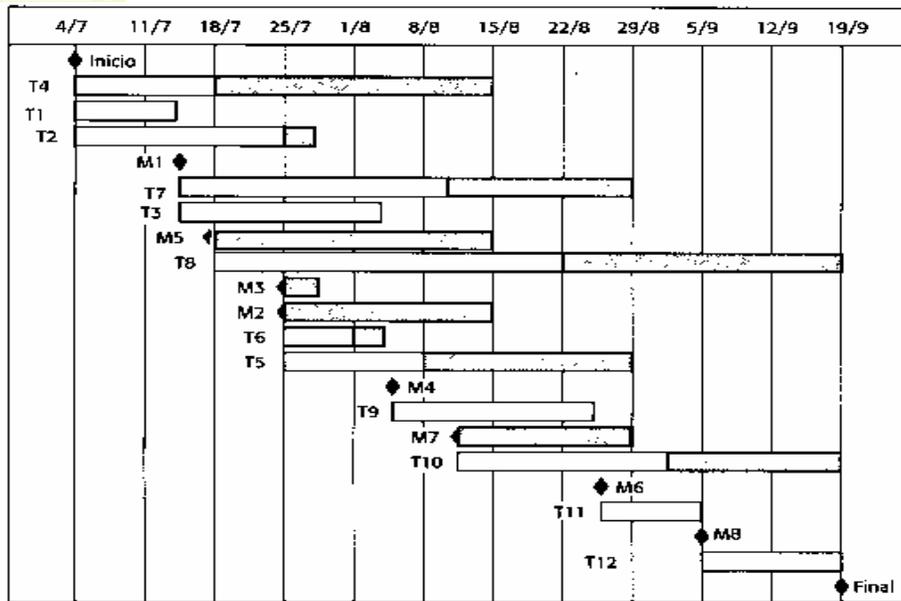


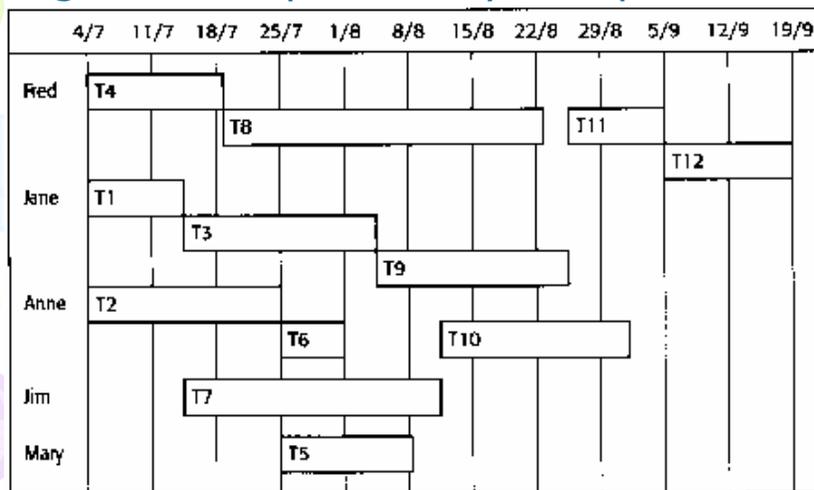
Gráfico de barras de las actividades

## Tabla de Asignación de personal

Tareas	Ingeniero
A1	Jane
A2	Anne
A3	Jane
A4	Fred
A5	Mary
A6	Anne
A7	Jim
A8	Fred
A9	Jane
A10	Anne
A11	Fred
A12	Fred

## Gráfico de Barra

- Asignación al personal y tiempos.



## Ejercicio

- La siguiente figura muestra un conjunto de actividades, duraciones y dependencias.
  - Diseñe una red de actividades y un gráfico de barras que muestren la programación de proyectos.

Sommerville, I., "Ingeniería de Software", 6ª ed., Addison Wesley

TAREA	DURACIÓN (DÍAS)	DEPENDENCIAS
T1	10	
T2	15	T1
T3	10	T1, T2
T4	20	
T5	10	
T6	15	T3, T4
T7	20	T3
T8	35	T7
T9	15	T6
T10	5	T5, T9
T11	10	T9
T12	20	T10
T13	35	T3, T4
T14	10	T8, T9
T15	20	T12, T14
T16	10	T15

## Relación de estimación de esfuerzo y duración de actividades

- Teniendo la estimación del esfuerzo total del proyecto, se debe repartir entre las diversas actividades
- Puede hacerse de proyectos anteriores
- Si no se tienen proyectos anteriores, considerar información publicada

## Reparto del esfuerzo por actividad

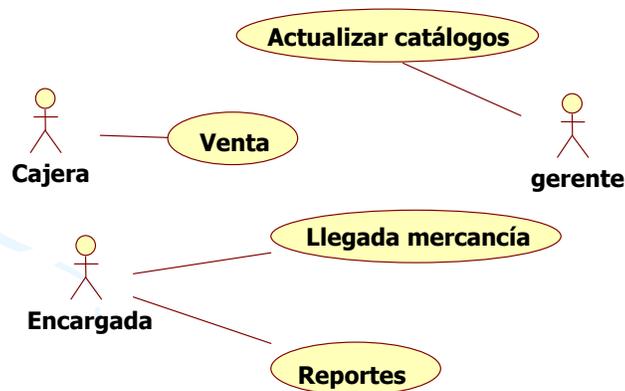
Actividad	Longstreet	Otros Promedios	RUP
Requerimientos	20%	20%	12%
Diseño	40%	25%	16%
Codificación	15%	23%	54%
Pruebas	25%	27%	18%

## Reparto del esfuerzo por actividad (2)

Actividad	Porcentaje
Desarrollo del proyecto	75%
Administración	15%
Otras	10%

## Ejemplo: Grupo misceláneas

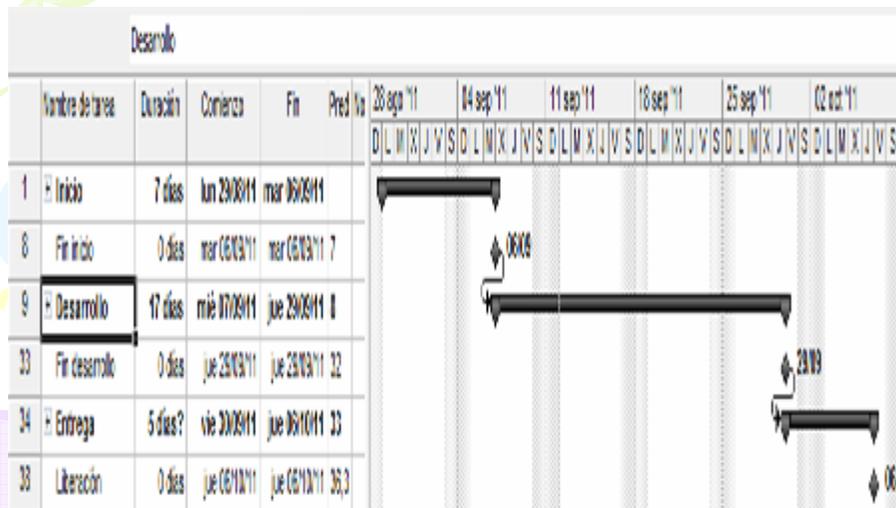
Se busca realizar sistema de punto de venta de las misceláneas, que tendrán una base de datos centralizada. Se estima una duración de seis semanas.



## Organización iteraciones

- Primera:
  - Terminar prototipo rápido
  - Crear base de datos, CU actualizar catálogos
- Segunda:
  - Venta típica y su prueba
- Tercera:
  - Venta casos especiales y llegada mercancía
- Cuarta:
  - Reportes

## Plan general



## Red de actividades (Inicio)



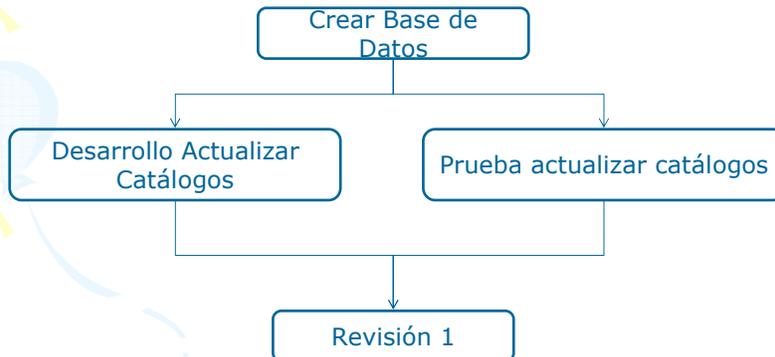
## Detalle Inicio

Inicio						
	Nombre de tarea	Duración	Comienzo	Fin	Pred No	
1	Inicio	7 días	lun 29/08/11	mar 06/09/11		
2	Plática inicial	0.5 días	lun 29/08/11	lun 29/08/11		
3	entrevistas us	1.5 días	lun 29/08/11	mar 30/08/11	2	
4	preparar mode	2 días	mié 31/08/11	jue 01/09/11	3	
5	establecer ver	2 días	mié 31/08/11	jue 01/09/11	3	
6	Revisar con cl	1 día	vie 02/09/11	vie 02/09/11	4,5	
7	terminar espec	2 días	lun 05/09/11	mar 06/09/11	6	
8	Fin inicio	0 días	mar 06/09/11	mar 06/09/11	7	

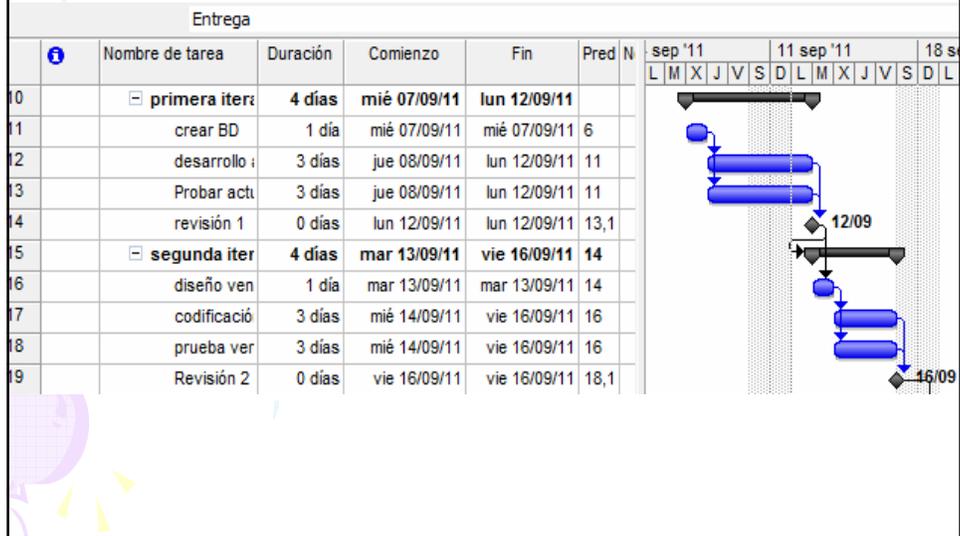
# Detalle Desarrollo

Desarrollo						sep '11	11 sep '11	18 sep '11	25 sep '11	02																
Nombre de tarea	Duración	Comienzo	Fin	Pred	No	L	M	X	J	V	S	D	L	M	X	J	V	S	D	L	M	X	J	V	S	D
1	+	Inicio	7 días	lun 29/08/11	mar 06/09/11																					
8		Fin inicio	0 días	mar 06/09/11	mar 06/09/11	7																				
9	-	Desarrollo	17 días	mié 07/09/11	jue 29/09/11	8																				
10	+	primera iter:	4 días	mié 07/09/11	lun 12/09/11																					
15	+	segunda iter	4 días	mar 13/09/11	vie 16/09/11	14																				
20	+	tercera iter	4 días	lun 19/09/11	jue 22/09/11	19																				
25	+	cuarta iterac	5 días	vie 23/09/11	jue 29/09/11	24																				
33		Fin desarrollo	0 días	jue 29/09/11	jue 29/09/11	32																				

# Red de actividades (Iteración 1)

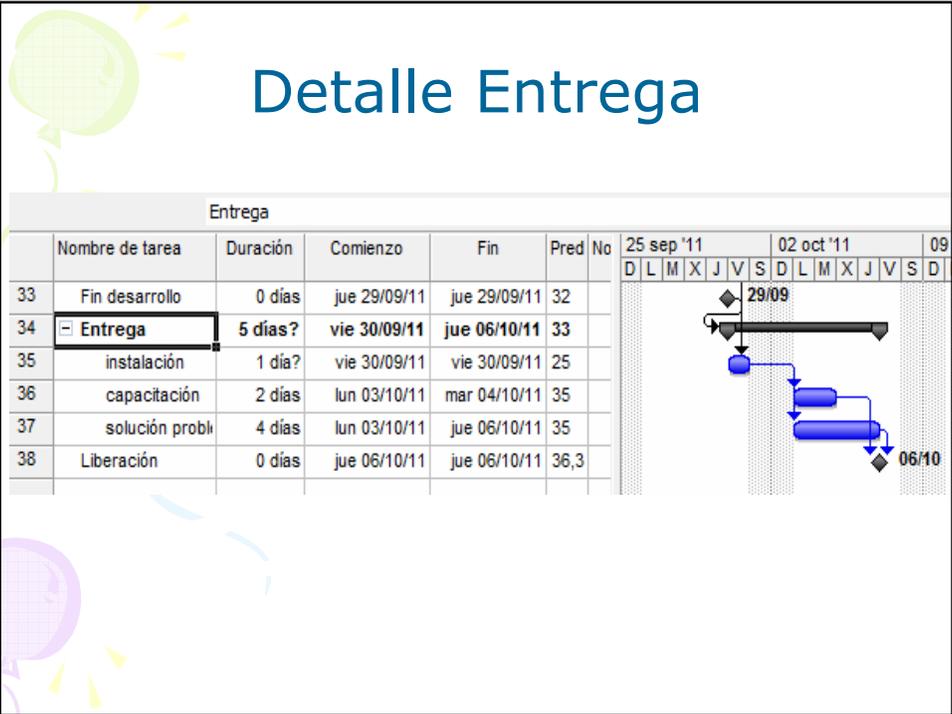
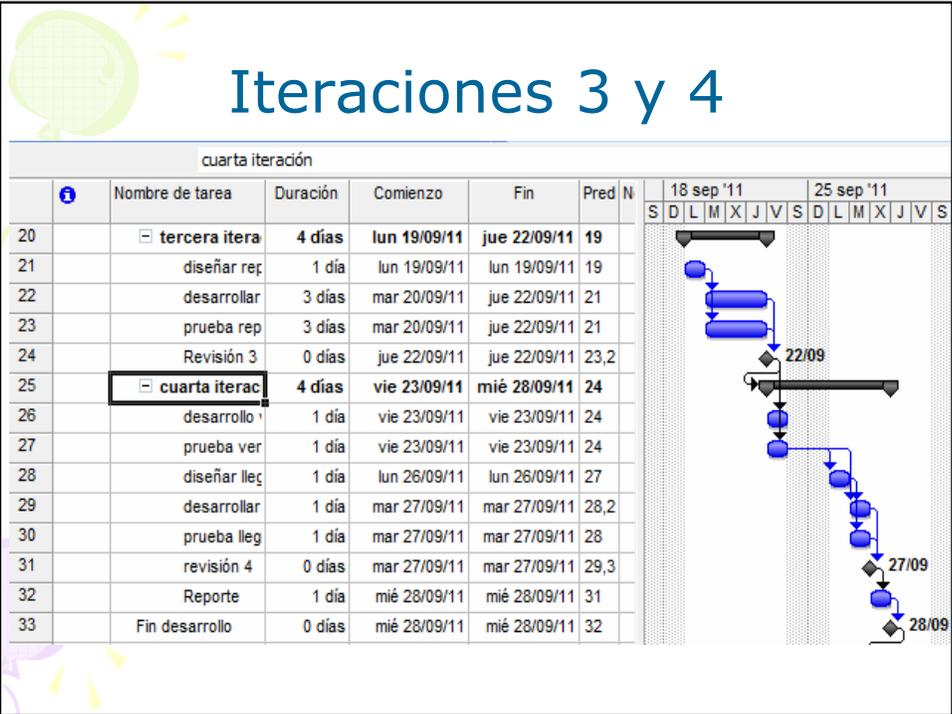


# Iteraciones 1 y 2



# Red de actividades (Iteración 4)





## Resumen del Esfuerzo

Actividad	Subactividad	Esfuerzo* (días-persona)
Inicio		9
Desarrollo		26
	Iteración 1	7
	Iteración 2	7
	Iteración 3	7
	Iteración 4	5
Entrega		7
<b>TOTAL</b>		<b>42</b>

\*Note que esta tabla marca los días persona, que difieren de los días calendario

## Ruta crítica en el ejemplo

Plática inicial	0.5
Entrevistas	1.5
Preparar modelos	2
Revisar	1
Terminar especificación	2
Crear BD	1
Catálogos	3
Diseño ventas	1
Ventas	3
Diseño llegadas	1
Codificación	1
Instalar	1
Resolver problemas	4
<b>TOTAL</b>	<b>27</b>

Quedan fuera actividades paralelas iguales o menores