

# Introducción a la Informática

## Ingeniería en Informática

Departamentos de Automática y Ciencias de la Computación

Tema 8

## Ingeniería del Software



Jaime Oyarzo Espinosa [jaime.oyarzo@uah.es](mailto:jaime.oyarzo@uah.es)  
Profesor Asociado, Universidad de Alcalá  
eLearning Consultant, LUND University, Sweden



## Introducción a los Sistemas de Información

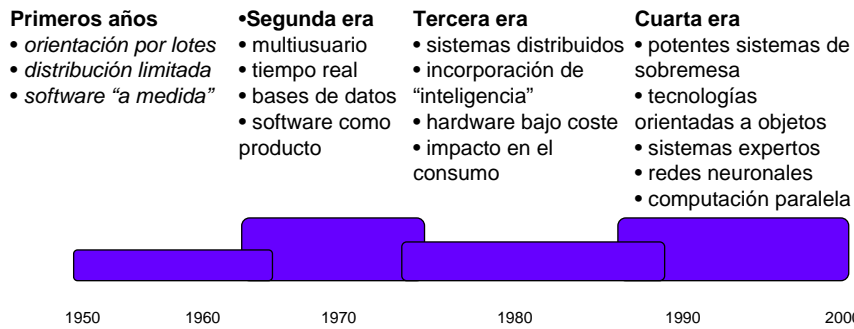
- El concepto de Sistema.
- El concepto de Información.
- Sistemas de Información.
  - Definición
  - Elementos
  - Estructura
- Aplicación de las T.I. a los S.I.

Tema 8

# Ingeniería del Software

## Introducción




### Evolución del Software



## ¿Cómo es el software?

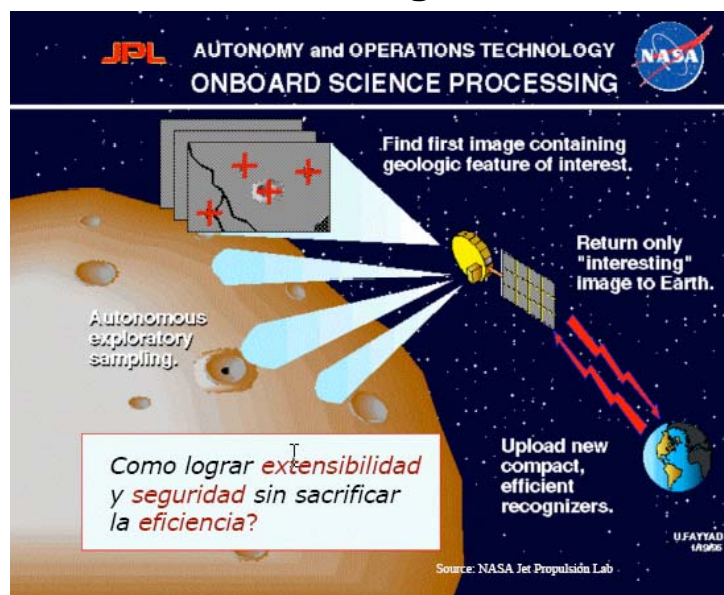
- Se desarrolla no se fabrica en el sentido clásico
- No se estropea
- Es complejo
- Es un "diferenciador"
- Parecido a "viejas fábricas"

## Demandas comerciales

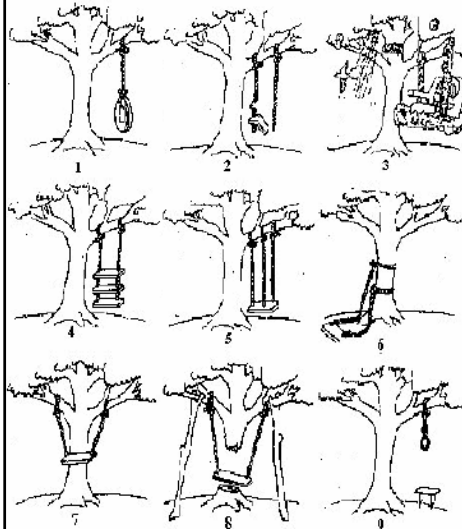
- Systems that must never crash and must always meet their deadlines. 
- Systems that must be secure, trustworthy, lightweight, and extensible. 
- Performance-critical open networking systems that are costly to shut down. 

Tema 8

## Software de alta seguridad

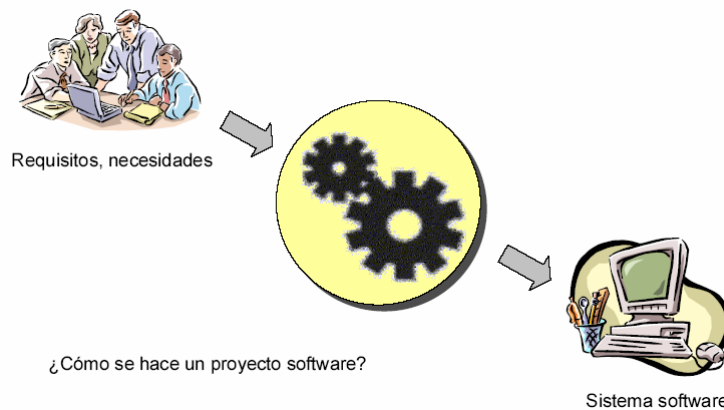


## Típico problema de incomunicación



1. **Necesidad:** lo que el cliente realmente quería.
2. **Cliente:** lo que fue capaz de describir como una clara necesidad.
3. **Proceso de ventas:** lo que el fabricante de software prometió al cliente.
4. **Requisitos:** los requisitos descritos por el cliente, tal como finalmente fueron entendidos.
5. **Análisis:** la especificación formal de los requisitos realizada por los analistas.
6. **Diseño:** la especificación del funcionamiento del sistema para satisfacer los requisitos analizados.
7. **Codificación:** lo que escribió el programador.
8. **Instalación:** lo que realmente fue instalado al cliente.
9. **Pruebas:** lo que los responsables vieron en el sistema.

## Proceso de desarrollo del Software



## **Evolución histórica y desarrollo de los S.I.**

- Antecedentes de la Informática.
- Evolución en el Hardware.
- Evolución en el Software.
- Evolución en el almacenamiento de datos.

Tema 8

## **Ingeniería del Software**

- Introducción
- Ciclo de vida del Software
- Metodologías
- La automatización (CASE)
- Garantía de Calidad (SQA)

Tema 8

## Re-Ingeniería del Software

- Re-Ingeniería del Software
  - Definiciones
  - Importancia de la Re-Ingeniería
  - Re-Ingeniería y CASE
  - Pasos de la Re-Ingeniería
- Reutilización del Software
  - Definición y Nueva Concepción
  - Ejemplos
  - Ventajas
  - Problemas

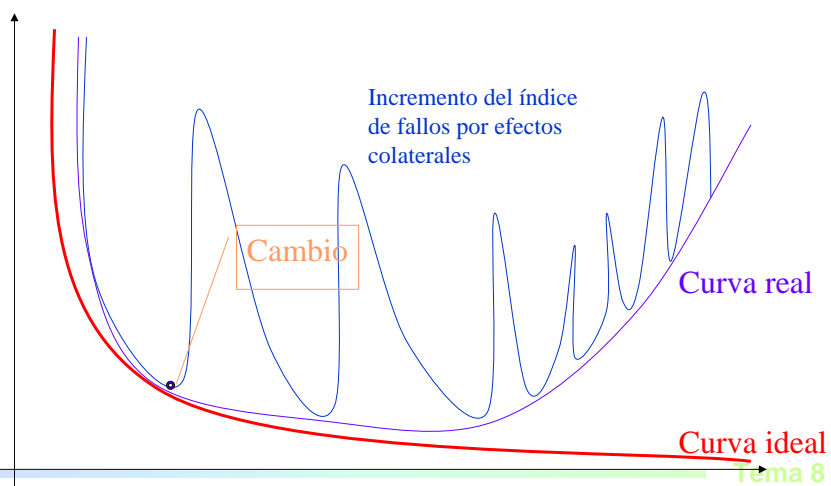
Tema 8

## Ingeniería del Software Introducción

- La importancia del Software
  - Tres primeras décadas : desafío en el desarrollo del Hard.
  - Durante los 80 : avances en microelectrónica que dan como resultado mayor potencia de cálculo y reducción de costes.
  - El problema hoy es diferente : mejorar la Calidad de las soluciones basadas en ordenador:
    - ✓ Soluciones implementadas con Software

Tema 8

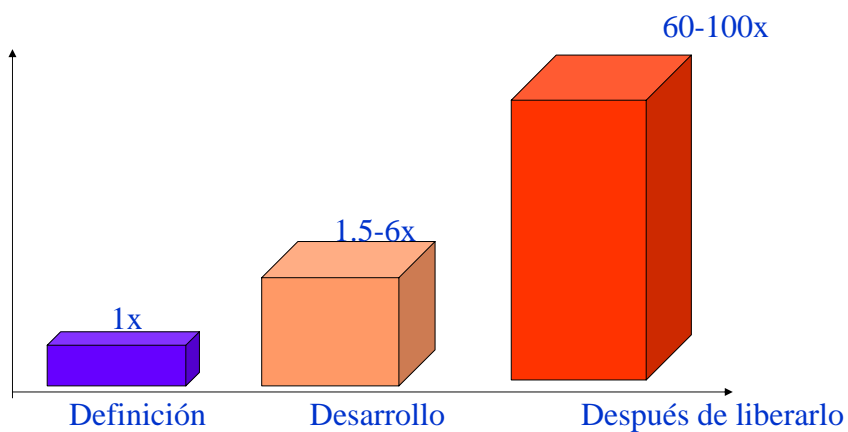
## Curva real de fallos del software



© JAGM/JOE

13

## El Coste del Cambio

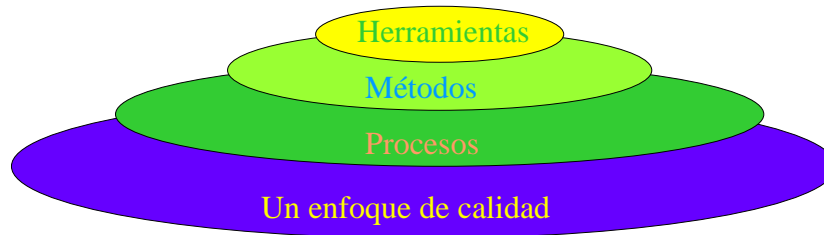


© JAGM/JOE

14

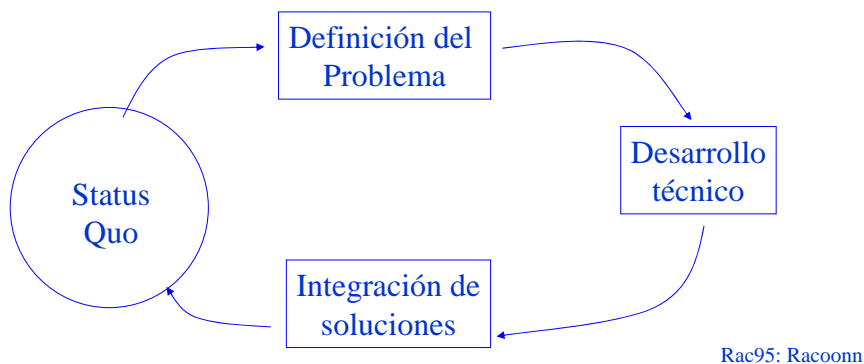
Tema 8

## Capas en la Ingeniería del Software



Tema 8

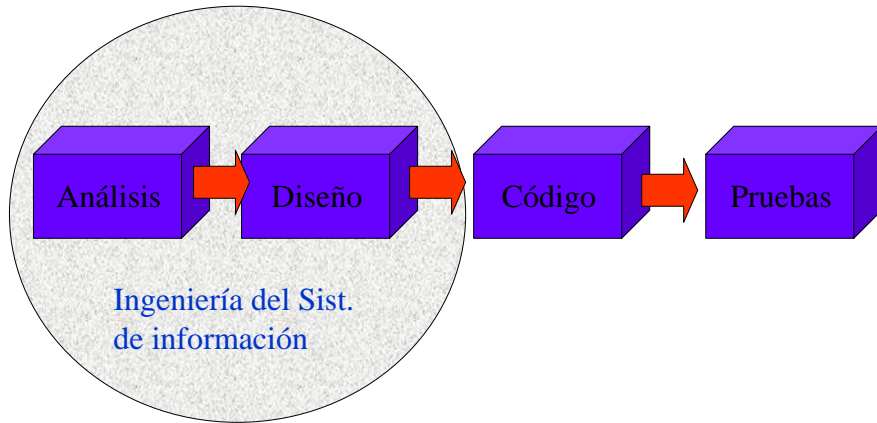
## Fases de un bucle de resolución de problemas



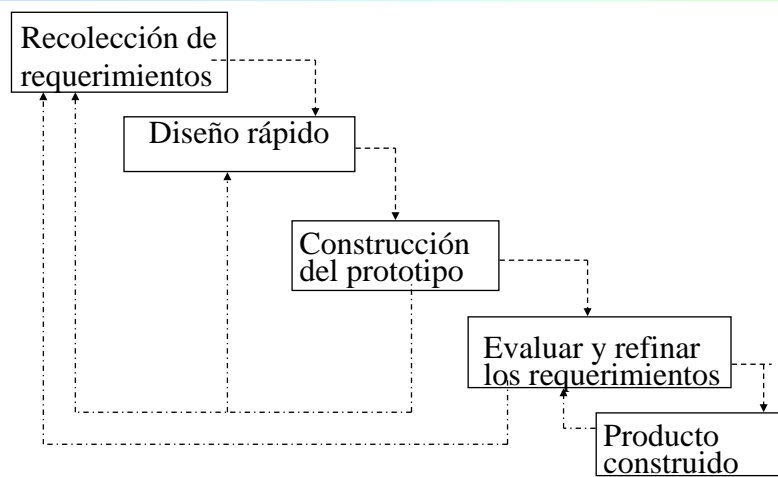
Tema 8



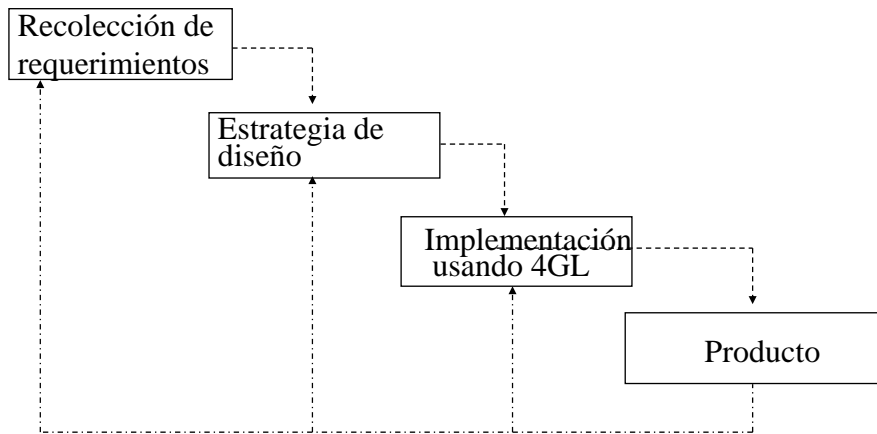
## El modelo lineal



## Construcción de prototipos



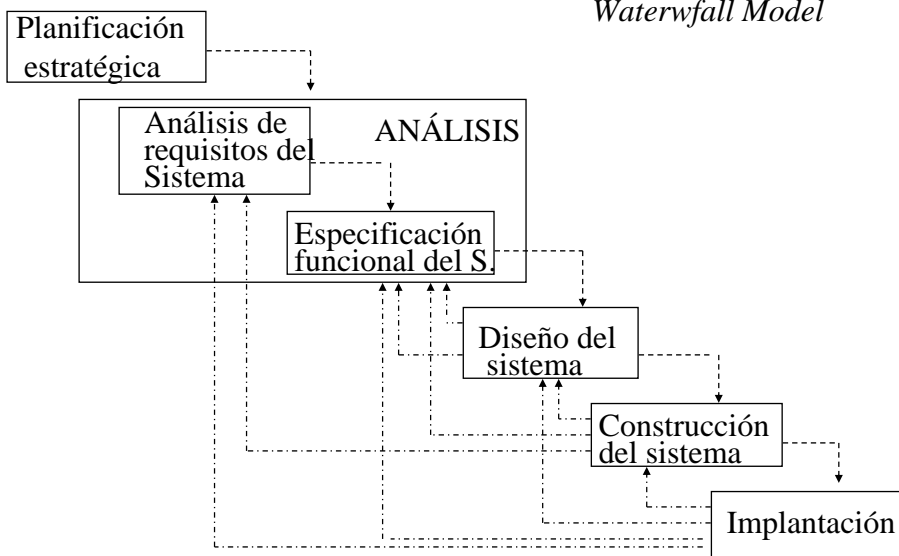
## Técnicas de cuarta generación



Tema 8

## El ciclo de vida clásico

*Waterfall Model*



## Ingeniería del Software Introducción

- La evolución del Software (2)
  - Sofisticación del hardware → desfase en la construcción de software para explotar su potencial .
  - No existe capacidad para abastecer la demanda de nuevos programas.
  - La capacidad de mantener lo existente está amenazada por el mal diseño y el uso de recursos inadecuados.

Tema 8

## Ingeniería del Software Principales problemas

- La planificación y la estimación de costes son frecuentemente imprecisas.
- La productividad no se corresponde con la demanda de servicios.
- La calidad del Software no llega a ser a veces ni aceptable.

Tema 8

## Ingeniería del Software

### Una definición

- Disciplina que se ocupa del establecimiento y uso de principios firmes de ingeniería, para obtener software económico que sea fiable y funcione eficientemente en máquinas reales [ Fritz Bauer].
- Tres elementos claves:
  - Métodos
  - Herramientas
  - Procedimientos(denominados frecuentemente:  
paradigmas de la ingeniería del software)

## Ingeniería del Software

### Los métodos

- Indican “como” construir el software
- Abarcan un amplio espectro:
  - planificación y estimación de proyectos
  - análisis de los requisitos del sistema y del software
  - diseño de estructuras de datos
  - arquitectura de programas y procedimientos algorítmicos
  - codificación, prueba y mantenimiento

## Ingeniería del Software

### Las herramientas

- Suministran un soporte automático para los métodos
- Existen herramientas para soportar cada uno de los métodos mencionados anteriormente
- Un entorno con integración de diferentes herramientas. Se denomina un sistema CASE (Ingeniería del Software Asistida por Ordenador)

Tema 8

## Ingeniería del Software

### Los procedimientos

- “Pegamento” que junta los métodos y las herramientas
- Definen:
  - la secuencia en la que se aplican los métodos
  - las entregas (documentos, informes, etc..) requeridas
  - los controles: ayudan a asegurar la calidad y coordinar los cambios
  - Directrices: ayudan a los gestores del software a evaluar el progreso

Tema 8

## Concepto de Ciclo de Vida

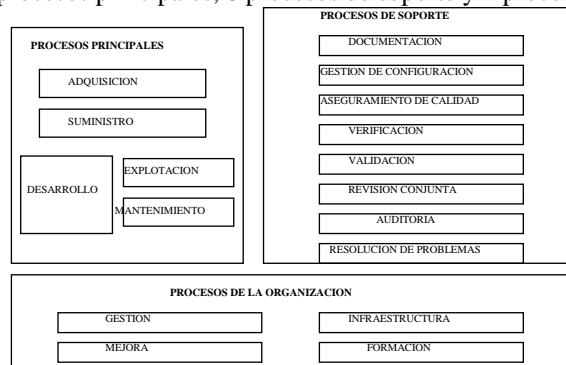
- Un gran problema de los S.I. es definir un marco de referencia común que:
  - pueda ser empleado por todos
  - en el que se definan procesos, actividades y tareas
- IEEE 1991- Standard for Developing Software Life Cycle Processes. ::> Aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software.
- ISO 1994 - Software life-cycle process ::> Marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso.

Tema 8

## Ciclo de Vida del Software

### ■ Procesos del Ciclo de Vida

Norma ISO 12207-1 Actividades del Ciclo de Vida se agrupan en 5 procesos principales, 8 procesos de soporte y 4 procesos generales.

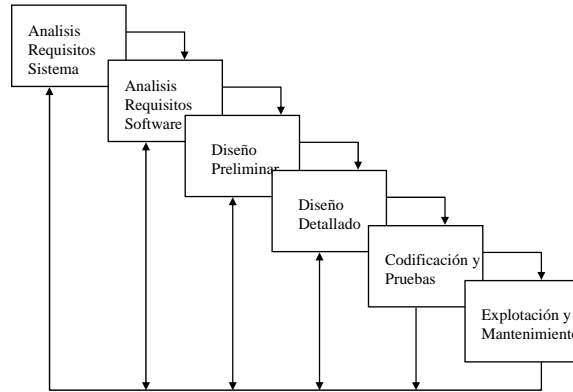


Tema 8

## Ciclo de Vida del Software

### Modelo en Cascada (Waterfall)

- [ROYCE-70][BOEHM-81][SOMMERVILLE-85][SIGWART-90]

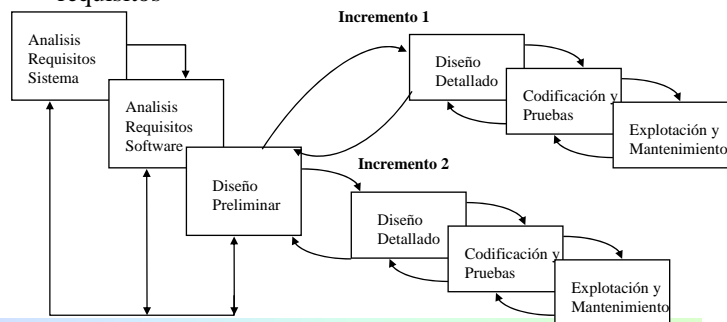


Tema 8

## Ciclo de Vida del Software

### Modelo Incremental [LEHMAN-84]

- Crear sistema añadiendo componentes funcionales (incrementos)
- En cada paso se actualiza el sistema con nuevas funcionalidades y requisitos



Tema 8

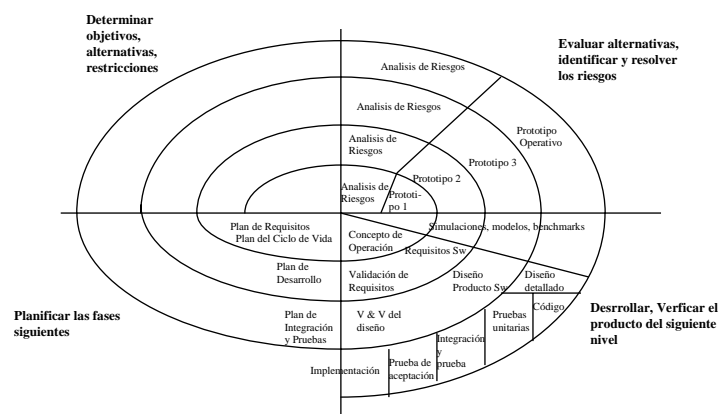
## Ciclo de Vida del Software

### Modelo en Espiral [BOEHM-88]

- Consta de una serie de ciclos
- Cada uno empieza identificando los objetivos, las alternativas y las restricciones del ciclo
- Una vez evaluadas las alternativas respecto a los objetivos y teniendo en cuenta las restricciones, se lleva a cabo el ciclo correspondiente para, una vez finalizado empezar a plantear el próximo.

## Ciclo de Vida del Software

### Modelo en Espiral [BOEHM-88]





## Modelos para desarrollos O.O.

- La programación O.O. plantea el desarrollo de sistemas de una manera iterativa e incremental.
- El intento de que el código sea reutilizable incrementa los costes de desarrollo entre un 10% y un 50%, por lo que es imprescindible un desarrollo que optimice esta inversión.
- Los modelos clásicos no permitían afrontar este tipo de programación.
- Modelo de Agrupamiento, Modelo fuente, Modelo Remolino, Modelo Pinball.

Tema 8

## Metodologías de desarrollo del Software Conceptos Generales

- “Conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de S.I.” [MADDISON-83]
- Con una metodología se intentan cubrir tres necesidades principales:
  - Mejores aplicaciones
  - Un proceso de desarrollo que identifica productos intermedios de cada fase de forma que se pueda planificar y controlar el proyecto
  - Un proceso estándar en la organización

Tema 8

## Metodologías de desarrollo del Software

### Características Principales

- Existencia de reglas predefinidas
- Cobertura total del ciclo de desarrollo
- Verificaciones intermedias
- Planificación y Control
- Comunicación efectiva
- Utilización sobre un abanico amplio de proyectos
- Fácil formación
- Herramientas CASE
- Debe contener actividades que mejoren el proceso de desarrollo
- Soporte al mantenimiento
- Soporte de la reutilización de software

Tema 8

## Metodologías de desarrollo del Software

### Metodologías Estructuradas

Creación de modelos (procesos, flujos, datos) de una manera descendente (top-down). Esta visión se puede enfocar en los procesos, en lo datos o en ambos:

- Metodologías orientadas a procesos
- Metodologías orientadas a datos jerárquicos
- Metodologías orientadas a datos no jerárquicos
- Mixtas

Tema 8

## Ejemplo de programación convencional

```
10 CLS
20 A=10
30 INPUT B
40 IF B=A THEN GOTO 50 ELSE GOTO 70
50 PRINT "A Y B SON IGUALES"
60 GOTO 100
70 IF A>B THEN GOTO 80 ELSE GOTO 90
80 B= B + 1; GOTO 40
90 B= B - 1; GOTO 40
100 END
```

## Ejemplo de programación Estructurada

```
PROGRAM NUMEROSIGUALES
BEGIN
  CLEARSCREEN;
  A :=10 ;
  INPUT B;
  REPEAT
    IF B=A THEN PRINT "A Y B SON IGUALES"
    ELSE REDUCEDIFERENCIA(A,B);
  UNTIL B=A;
END;

PROCEDURE REDUCEDIFENCIA(A,B);
BEGIN
  IF A>B THEN B:= B+1
  ELSE B:= B - 1
END
```

## ***Metodologías Estructuradas Orientadas a Procesos***

- Fundadas sobre el modelo básico entrada/proceso/salida.
- Se enfocan en la parte del proceso
- Autores: DEMARCO, GANE&SARSON, YOURDON
- Especificación estructurada basada en:
  - Diagramas de flujo de datos (DFD)
  - Diccionario de Datos
  - Especificaciones de Procesos

Tema 8

## ***Metodologías Estructuradas Orientadas a Datos Jerárquicos***

- Fundadas sobre el modelo básico entrada/proceso/salida.
- Se enfocan en la parte del proceso
- Autores: JACKSON, CAMERON, WARNIER
- Se definen las estructuras de datos y a partir de estas se derivan los componentes procedimentales:
  - la estructura de control de programa debe ser jerárquica
  - se deriva de la estructura de datos
  - el diseño lógico debe preceder y estar separado del físico

Tema 8

## **Metodologías Estructuradas Orientadas a Datos No Jerárquicos**

- Los datos son el corazón del S.I.
- El modelo esta formado por el conjunto de entidades básicas y las interrelaciones entre ellas
- Autor: MARTIN y FINKELSTEIN
- Cuatro etapas:
  - Planificación
  - Análisis
  - Diseño
  - Construcción

Tema 8

## **Metodologías de desarrollo del Software Metodologías Orientadas a Objetos**

- En O.O. se examina el dominio del problema como un conjunto de objetos que interactúan entre sí
- En las tradicionales : dicotomía entre funciones que llevan a cabo los programas y datos que se almacenan en bases de datos
- La O.O. propugna un enfoque unificador
- Dos enfoques:
  - Puros : [OOD-BOOCH][WIRFS&BROCK]
  - Evolutivos : [OMT-RUMBAUGH] [MARTIN&ODELL]
- Segunda generación: “METODO UNIFICADO-BOOCH&RUMBAUGH”

Tema 8

## Metodologías de desarrollo del Software

- Principales Metodologías “Oficiales”
- Metodología MERISE
- Metodología SSADM
- Metodología MÉTRICA
  - Estructurada en fases, módulos, actividades y tareas
  - FASE 0 : Plan de Sistemas de Información
  - FASE 1 : Análisis de Sistemas
  - FASE 2 : Diseño de Sistemas
  - FASE 3 : Construcción de Sistemas
  - FASE 4 : Implantación de Sistemas

Tema 8

## La automatización del Software (CASE)

- Clasificación del CASE
  - CASE superior (Upper CASE) : concierne a la fase de planificación y gestión del proyecto.
  - CASE intermedio (Middle CASE) : referidas a la automatización del Análisis y Diseño de S.I.
  - CASE inferior (Lower CASE) : pretende la asistencia del ordenador en la construcción (programación o generación de código) de los S.I.

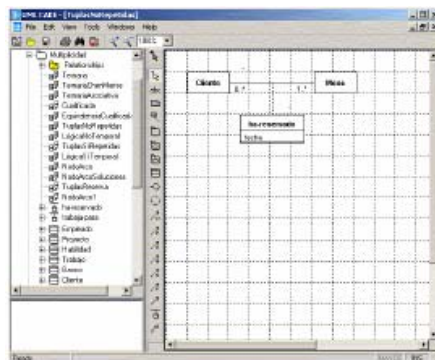
Tema 8

## La automatización del Software (CASE) *Tipos de Herramientas*

- Diagramación (dibujo de diagramas estructurados y creación de especificaciones gráficas)
- Generadores de Pantallas e Informes (prototipado de la interfase de usuario, Repositorios e Informes)
- Verificación y Análisis
- Generadores de Código
- Mantenimiento
- Planificación

Tema 8

## Herramientas de modelado CASE



Tema 8

## Ejemplo: símbolos Oracle

ORACLE SYMBOL	ORACLE CARDINALITY NOTATION	CARDINALITY DESCRIPTION IN SIMPLE WORDS
	M:1 (M to 0)	Mandatory Many to Optional One
	M:1 (0 to 0)	Optional Many to Optional One
	M:1 (0 to M)	Optional Many to Mandatory One
	M:1 (M to M)	Mandatory Many to Mandatory One
	M:M (M to 0)	Mandatory Many to Optional Many
	M:M (0 to 0)	Optional Many to Optional Many
	1:1 (M to 0)	Mandatory One to Optional One
	1:1 (0 to 0)	Optional One to Optional One
	1:1 (M to M)	Mandatory One to Mandatory One

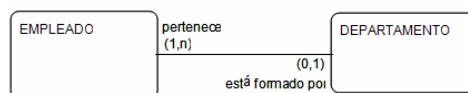
Tema 8

© JAGM/JOE

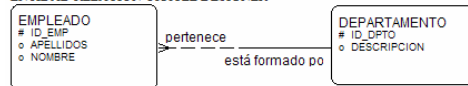
47

## Ejemplo: símbolos Oracle

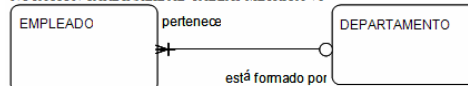
ENTIDAD RELACION



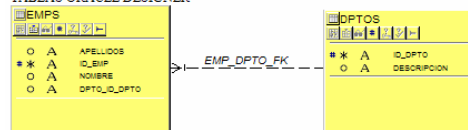
ENTIDAD RELACION ORACLE DESIGNER



NOTACION CARDINALIDAD TABLAS METRICA V3



TABLAS ORACLE DESIGNER





## Garantía de Calidad del Software (SQA)

### Consideraciones previas

- Principal fin en la Ingeniería del Software: producir software de gran calidad
- SQA: “actividad de protección” que se aplica a lo largo de todo el proceso de ingeniería del software.
- La SQA engloba:
  - métodos y herramientas de análisis, diseño, codificación y prueba.
  - revisiones técnicas formales que se aplican en cada paso.
  - estrategia de prueba.
  - control de la documentación del software y de los cambios realizados.
  - procedimiento que asegure un ajuste a los estándares de desarrollo.
  - mecanismos de medida y de información.

Tema 8

## Garantía de Calidad del Software (SQA) Definición

- Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Tema 8

## Garantía de Calidad del Software (SQA) Tres puntos importantes:

- Los requisitos del software son la base de las medidas de calidad.
- Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software.
- La calidad del software queda en entredicho si no se ajusta a requerimientos implícitos (un buen mantenimiento, etc..).

## Garantía de Calidad del Software (SQA) *Factores de Calidad del Software.*

Los factores que afectan a la calidad del software se pueden clasificar en dos grandes grupos:

- factores que pueden ser medidos directamente (errores/ KLDC/ unidad de tiempo).
- factores que solo pueden ser leídos indirectamente (facilidad de uso, etc..)

En ambos casos debemos comparar el software con alguna referencia y llegar a una indicación de calidad.

## Referencias Internet

- [Welcome to McGraw-Hill's Online Learning Center](#)
- IBM-Rational  
<http://www-106.ibm.com/developerworks/rational/newto/#4>
- General
- Escuela Técnica Superior de Ingeniería Informática – UAH  
<http://www.etsii.uah.es/>
- Departamento de Ciencias de la Computación  
[www.cc.uah.es](http://www.cc.uah.es)

Tema 8

## Bibliografía

- J.M. MARTÍNEZ, L. LÓPEZ, B. CASTAÑO, J.A. MALPICA, J.R. HILERA y J.A. GUTIÉRREZ. *Metodología de Desarrollo de Sistemas de Información*. Ed. Universidad de Alcalá, 1995.
- GANE, C. & SARSON, T. *Análisis Estructurado de Sistemas*. Ed. El Ateneo, 1990.
- HAWRYSZKIEWYCZ, I.T. *Introducción al Análisis y Diseño de Sistemas*. Ed. Anaya, 1990.
- KENDALL, K.E. & KENDALL, J.E. *Análisis y Diseño de Sistemas*. Ed. Prentice-Hall, 1995.
- *Metodología de Planificación y Desarrollo de Sistemas de Información: METRICA Version 3.0*. Ed. Ministerio para las Administraciones Públicas, 2001.

Tema 8