



PROGRAMA DE ASIGNATURA		2003/2004	
ASIGNATURA	Ingeniería del software I	CÓDIGO	402
CICLO	Segundo	PLAN DE ESTUDIOS	1994
ESPECIALIDAD		CURSO	4º
CRÉDITOS	12	CENTRO	Facultad
CARÁCTER	Anual	CURSO ACADÉMICO	2003/2004

## I. TEMARIO

### 1ª Parte .- El producto y el proceso.

#### Capítulo 1.- El producto.

- 1.1. La evolución del software.
- 1.2. El software.
- 1.3. Software : una crisis en el horizonte.
- 1.4. Mitos del software.
- 1.5. Resumen.

#### Capítulo 2.- El proceso.

- 2.1. Ingeniería del software : una tecnología estratificada.
- 2.2. El proceso del software.
- 2.3. Modelos de proceso de software.
- 2.4. El modelo lineal secuencial.
- 2.5. El modelo de construcción de prototipos.
- 2.6. El modelo DRA.
- 2.7. Modelos de procesos evolutivos de software.
- 2.8. El modelo de métodos formales.
- 2.9. Técnicas de cuarta generación.
- 2.10. Tecnología de procesos.
- 2.11. Producto y proceso.
- 2.12. Resumen.

### 2ª Parte : Análisis estructurado: Herramientas de modelado.

#### Capítulo 3.- Diagramas de flujo de datos.

- 3.1. Los componentes de DFD.
- 3.2. Guía para la construcción de un DFD.
- 3.3. DFD por niveles.
- 3.4. Extensiones del DFD para sistemas de tiempo real.

#### Capítulo 4.- El diccionario de datos.

- 4.1. La necesidad de la notación en el diccionario de datos.
- 4.2. Notación del diccionario de datos.



4.3 Como mostrar el diccionario de datos al usuario.

4.4. Implantación del diccionario de datos.

#### Capítulo 5.- Especificaciones de proceso.

5.1. Lenguaje estructurado.

5.2. Pre/post condiciones.

5.3. Tablas de decisión.

5.4. Otras herramientas de especificación de proceso.

#### Capítulo 6.- Diagramas de entidad-relación.

6.1. Los componentes de un DER.

6.2. Reglas para la construcción de diagramas de entidad-relación.

6.3. Extensiones al diccionario de datos para diagramas E-R.

#### Capítulo 7.- Diagramas de transición de estados.

7.1. Notación de los diagramas de transición de estados.

7.2. Diagramas particionados.

7.3. Construcción del diagrama de transición de estados.

7.4. La relación del DTE con los demás componentes del modelo.

#### Capítulo 8.- Balanceo de modelos.

8.1. Balanceo del DFD y el DD.

8.2. Balanceo del DFD y la especificación de proceso.

8.3. Balanceo de las especificaciones del proceso con el DFD y el DD.

8.4. Balanceo del diccionario de datos con el DFD y las especificaciones del proceso.

8.5. Balanceo del DER con el DFD y las especificaciones de proceso.

8.6. Balanceo del DFD y el diagrama de transición de estados.

### **3ª Parte : Análisis estructurado: El proceso de análisis.**

#### Capítulo 9.- El modelo esencial.

9.1. El enfoque del modelado clásico y por qué no funcionó.

9.2. El modelo esencial.

9.3. Qué hacer si se debe construir un modelo de implantación.

#### Capítulo 10- El modelo ambiental.

10.1. Herramientas usadas para definir el ambiente.

10.2. Construcción del modelo ambiental.



## Capítulo 11.- Construcción de un modelo preliminar de comportamiento.

- 11.1. El enfoque clásico.
- 11.2. Identificación de respuestas a acontecimientos.
- 11.3. Conexión de las respuestas a acontecimientos.
- 11.4. Desarrollo del modelo inicial de datos.

## Capítulo 12.- Terminado del modelo de comportamiento.

- 12.1. Terminado del modelo del proceso.
- 12.2. Terminado del modelo de datos.
- 12.3. Terminado del DTE.

## Capítulo 13.- El modelo de implantación del usuario.

- 13.1. Determinación de la frontera de automatización.
- 13.2. Determinación de la interfaz humana.
- 13.3. Identificación de las actividades de apoyo manual adicional.
- 13.4. Especificación de restricciones operacionales.

## **4ª Parte : Gestión de proyectos de software.**

### Capítulo 14.- Conceptos sobre gestión de proyectos.

- 14.1. El espectro de la gestión.
- 14.2. Personal.
- 14.3. El problema.
- 14.4. El proceso.
- 14.5. El proyecto.
- 14.6. Resumen.

### Capítulo 15.- El proceso de software y métricas del proyecto.

- 15.1. Medidas, métricas e indicadores.
- 15.2. Métricas en el proceso y dominios del proyecto.
- 15.3. Mediciones del software.
- 15.4. Reconciliación de los diferentes enfoques de métricas.
- 15.5. Métricas para la calidad del software.
- 15.6. Integración de las métricas dentro del proceso del software.
- 15.7. Resumen.

### Capítulo 16.- Planificación de proyectos de software.

- 16.1. Observaciones sobre la estimación.



- 16.2. Objetivos de la planificación del proyecto.
- 16.3. Ámbito del software.
- 16.4. Recursos.
- 16.5. Estimación del proyecto de software.
- 16.6. Técnicas de descomposición.
- 16.7. Modelos empíricos de estimación.
- 16.8. La decisión desarrollar-comprar.
- 16.9. Herramientas automáticas de estimación.
- 16.10. Resumen.

#### Capítulo 17.- Gestión del riesgo.

- 17.1. Estrategias de riesgo proactivas y reactivas.
- 17.2. Riesgos del software.
- 17.3. Identificación del riesgo.
- 17.4. Proyección del riesgo.
- 17.5. Reducción, supervisión y gestión del riesgo.
- 17.6. Riesgos y peligros para la seguridad.
- 17.7. El plan RSGR.
- 17.8. Resumen.

#### Capítulo 18.- Planificación temporal y seguimiento del proyecto.

- 18.1. Conceptos básicos.
- 18.2. La relación entre las personas y el esfuerzo.
- 18.3. Definición de un conjunto de tareas para el proyecto de software.
- 18.4. Selección de las tareas de ingeniería del software.
- 18.5. Refinamiento de las tareas principales.
- 18.6. Definir una red de tareas.
- 18.7. La planificación temporal.
- 18.8. El plan del proyecto.
- 18.9. Resumen

**Se realizarán sesiones prácticas en el laboratorio con herramienta interactiva de estimación, obtención de métricas, etc. para el desarrollo de proyectos de software.**

#### Capítulo 19.- Control de calidad del software.

- 19.1. Conceptos de calidad.
- 19.2. La tendencia de la calidad.
- 19.3. Garantía (aseguramiento) de calidad del software.
- 19.4. Revisiones del software.
- 19.5. Revisiones técnicas formales.
- 19.6. Enfoques formales a la SQA.



- 19.7. Garantía de calidad estadística.
- 19.8. Fiabilidad del software.
- 19.9. El plan de SQA.
- 19.10. Los estándares de calidad ISO 9000.
- 19.11. Resumen.

## Capítulo 20.- Gestión de la configuración del software.

- 20.1. Gestión de la configuración del software.
- 20.2. El proceso de GCS.
- 20.3. Identificación de objetos en la configuración del software.
- 20.4. Control de versiones.
- 20.5. Control de cambios.
- 20.6. Auditoría de la configuración.
- 20.7. Informes de estado.
- 20.8. Estándares de GCS.
- 20.9. Resumen.

## **5ª Parte : Métodos convencionales de la ingeniería del software.**

### Capítulo 21.- Métodos de prueba del software.

- 21.1. Fundamentos de la prueba del software.
- 21.2. Diseño de casos de prueba.
- 21.3. Prueba de caja blanca.
- 21.4. Prueba del camino básico.
- 21.5. Prueba de la estructura de control
- 21.6. Prueba de caja negra.
- 21.7. Prueba de entornos y aplicaciones especializadas.
- 21.8. Resumen.

### Capítulo 22.- Estrategias de prueba del software.

- 22.1. Un enfoque estratégico para la prueba del software.
- 22.2. Aspectos estratégicos.
- 22.3. Prueba de unidad.
- 22.4. Prueba de integración.
- 22.5. Prueba de validación.
- 22.6. Prueba del sistema.
- 22.7. El arte de la depuración.
- 22.8. Resumen.

## **6ª Parte : Temas avanzados de ingeniería del software.**

### Capítulo 23.- Reutilización del software.



- 23.1. Asuntos de gestión.
- 23.2. El proceso de reutilización.
- 23.3. Ingeniería de dominios.
- 23.4. Construcción de componentes reutilizables.
- 23.5. Clasificación y recuperación de componentes.
- 23.6. Economía de la reutilización del software.
- 23.7. Resumen.

#### Capítulo 24.- Reingeniería.

- 24.1. Reingeniería de procesos de negocios.
- 24.2. Reingeniería del software.
- 24.3. Ingeniería inversa.
- 24.4. Reestructuración.
- 24.5. Ingeniería progresiva (directa).
- 24.6. Economía de la reingeniería.
- 24.7. Resumen.

#### Capítulo 25.- Ingeniería del software cliente/servidor.

- 25.1. Estructura de los sistemas cliente/servidor.
- 25.2. Ingeniería del software para sistemas C/S.
- 25.3. Problemas de modelado de análisis.
- 25.4. Diseño para sistemas C/S.
- 25.5. Problemas de prueba.
- 25.6. Resumen.

#### Capítulo 26.- Plataforma Web.

#### Capítulo 27.- Ingeniería del software asistida por computadora.

- 27.1. ¿ Qué significa CASE ?.
- 27.2. Bloques básicos de CASE.
- 27.3. Una taxonomía de herramientas CASE.
- 27.4. Entornos CASE integrados.
- 27.5. La arquitectura de integración.
- 27.6. El depósito CASE.
- 27.7. Resumen.



## II. BIBLIOGRAFÍA.

### A. Básica

**ROGER S. PRESSMAN** : *Ingeniería del software. Un enfoque práctico (quinta edición)*. Mc Graw Hill.

**EDWARD YOURDON** : *Análisis estructurado moderno*. Prentice Hall.

### B. Complementaria y de consulta.

**RICHARD BARKER** : *El modelo entidad-relación CASE\*METHOD*. Adison Wesley / Diaz de Santos.

**CHRIS GANE - TRISH SARSON** : *Análisis estructurado de sistemas*. El Ateneo.

**JAMES A. SENN** : *Análisis y diseño de sistemas de información*. Mac Graw Hill.

## III. NORMAS ACADÉMICAS.

1ª - Se realizarán dos exámenes cuatrimestrales\_

- el 1º en Febrero, de la materia impartida hasta ese momento

- el 2º en Junio, de la materia impartida desde Febrero a Junio para los alumnos que hubiesen aprobado el 1º parcial; de la materia impartida durante todo el curso para los alumnos que hubiesen suspendido el 1º parcial.

2ª - Los alumnos que no aprobasen los 2 parciales en Junio deberán examinarse en Septiembre de la totalidad de la materia.

3ª - Cada alumno deberá realizar, y defender, un proyecto de ingeniería del software.

4ª - Para aprobar la asignatura cada alumno deberá aprobar los 2 parciales y el proyecto. La nota final se obtendrá como suma del 60% de la nota media de ambos parciales mas el 40% de la nota del proyecto.

Se podrán aprobar los 2 parciales en Junio y presentar el proyecto en Septiembre y viceversa.

Los alumnos que no superasen la asignatura en Septiembre deberán repetir la totalidad de la materia, y el proyecto, el curso siguiente, es decir, no se conservará ninguna nota de un curso para otro.