

1. OBJETIVOS:

Entender los conceptos teórico-prácticos que se emplean en la fase de diseño de un proyecto de software.

Entender las metodologías de diseño para las diferentes estrategias de desarrollo (estructurado, orientado a objetos, orientado a eventos, centrado en el usuario, centrado en la estructura de datos)

Conocer los distintos modelos de desarrollo y ciclos de vida.

Conocer los métodos ágiles.

Conocer los métodos formales.

Interiorizarse en los métodos, técnicas y herramientas que se usan en el diseño de software y como son usados para diseñar distintos tipos de sistemas.

Aprender a usar patrones de diseño.

2. CONTENIDOS:

A continuación se describe con mayor detalle cada una de tales unidades:

UNIDAD 1:

El proceso de diseño – Introducción

- a. Los conceptos de Ingeniería y Diseño
- b. Ingeniería de software y diseño de software en el marco de la Ingeniería de sistemas
- c. Conceptos de desarrollo y diseño de software (Repaso y presentación introductoria de conceptos básicos)
- d. Introducción al UML, vistas de UML, los modelos dinámicos y estáticos
- e. Introducción a las herramientas CASE

UNIDAD 2:

Diseño de software y su relación con las demás fases

- a. Relación a las fases en las que se:
 - I. definen estrategias (Planificación Estratégica)
 - II. diseñan los procesos de negocio (Análisis)
 - III. establecen los requerimientos (Análisis)
 - IV. definen las especificaciones (Análisis)
 - V. construye el código (Programación / Construcción)
 - VI. prueba el sistema (testing)

- VII. mantiene el sistema (Mantenimiento)
- b. Repaso de los modelos de desarrollo de software: cascada, evolutivo, espiral, iterativo, basado en componentes, ágil.
- c. El diseño en el UP (Proceso Unificado)
- d. Reingeniería e ingeniería inversa
- e. Los casos de uso y el modelo conceptual del sistema como medios para determinar las especificaciones funcionales del sistema

UNIDAD 3:

Metodologías ágiles

- a. Antecedentes.
- b. Manifiesto Ágil.
- c. XP (extremme programming),
- d. SCRUM,
- e. UP como método ágil.
- f. Artefactos de las metodologías ágiles
- g. Documentación en metodologías ágiles

UNIDAD 4:

Principios de buen diseño

- a. Principios del buen diseño:
 - i. abstracción
 - ii. encapsulamiento de la información
 - iii. cohesión
 - iv. acoplamiento
 - v. persistencia de datos
- b. Estrategias de diseño:
 - i. orientado a la funcionalidad,
 - ii. centrado en la estructura de los datos,
 - iii. orientado a objetos,
 - iv. orientado a eventos,
 - v. centrado en el usuario,
- c. Métodos formales de diseño.
- d. Niveles de diseño: diseño de la arquitectura, diseño de alto nivel y diseño detallado
- e. Atributos de calidad del software y su relación con el diseño
- f. Métricas de calidad del diseño

UNIDAD 5:

Diseño de la arquitectura de un sistema.

- a. Qué es la arquitectura de un sistema de software y cuál es su uso
- b. El modelo conceptual y la arquitectura del sistema
- c. Como influyen las características del sistema en su arquitectura
- d. El proceso de diseñar la arquitectura
- e. Diagramas UML de diseño arquitectónico
 - i. Diagramas de paquetes
 - ii. Diagramas de componentes
 - iii. Diagramas de despliegue
- f. Estilos de organización
- g. Estilos de control
- h. Comunicaciones entre procesos, programas y módulos.
- i. Arquitectura de sistemas distribuidos
- j. Arquitecturas de referencia, patrones de arquitectura
- k. Frameworks (marcos de trabajo)
- l. Cómo se documenta la Arquitectura de un Sistema

UNIDAD 6:

Metodología de diseño estructurado

- a. Concepto de módulo y de caja negra, medidas de cohesión y acoplamiento de módulos
- b. Heurísticas de diseño, tamaño del módulo, fan-in, fan-out, forma de mezquita
- c. Criterios para la organización: transformaciones, transacciones. procedimientos, dispositivos
- d. Diagramas
 - Diagrama de Estructura (Carta de Estructura)
 - El Diagrama de Flujo
 - Diagrama de Flujo de Datos (DFD)
 - Diagramas UML de actividad
- e. Patrones de diseño estructurado
- f. Como se documenta un diseño estructurado

UNIDAD 7:

Metodología de diseño orientado a objetos

- a. EL paradigma orientación a objetos y que es diseñar en este paradigma
- b. Como diseñar y diagramar: clases de alto nivel y de nivel detallado; diagramas de secuencia de alto nivel y de nivel detallado.
- c. Clases genéricas (templates)
- d. Patrones de diseño: factory method, abstract factory, prototype, adapter, decorator, composite, facade, proxy, chain of responsibility,

- command, state, iterator, mediator, strategy, visitor, singleton
- e. Diagramas UML de diseño OO:
 - i. Diagramas de clase
 - ii. Diagramas de secuencia
 - iii. Diagramas de actividad
- f. Generación de código
- g. Como se documenta un diseño orientado a objetos

UNIDAD 8:

Diseño de interfaces

- a. Las interfaces del usuario.
 - i. Prototipado
 - ii. Atributos
 - iii. Evaluación
- b. Interfaces entre sistemas
- c. Interfaces entre procesos
- d. Seguridad y control de tráfico de información.
- e. Como se documenta el diseño de una interface del usuario

UNIDAD 9:

Diseño del modelo de datos del sistema

- a. Persistencia de datos
- b. Diseño de Base de Datos
 - Las vistas lógicas y físicas de las bases de datos
- c. Diagrama de entidad – relación.
- d. Base de datos relacional
- e. Formas normales
- f. El idioma SQL
- g. Persistencia de objetos
- h. Base de datos orientada a objetos
- i. Utilización de herramientas CASE (ER-win)
- j. Como se documenta el diseño del modelo de datos

UNIDAD 10:

Diseño de sistemas orientados a eventos.

- a. Descripción de sistemas orientados a eventos
- b. Aplicado a diseño de la interfaz del usuario
- c. Aplicado a sistemas de tiempo real
- d. Aplicado a sistemas de producción basados en reglas
- e. Diagramas de transición de estados para gestionar eventos
- f. Especificidades en la documentación del diseño de sistemas

orientados a eventos

UNIDAD 11:

Diseño de sistemas de tiempo real

- a. Caracterización de los sistemas de tiempo real.
- b. Sistemas críticos
- c. El entorno de tiempo real
- d. Respuesta en tiempo real
- e. Consideraciones de arquitectura
- f. Recuperación de Fallas
- g. Comunicación en tiempo real - Comunicación asincrónica.
- h. Validación.
- i. Redes de Petri.
- j. Diagramas UML de diseño de sistemas de tiempo real
 - i. Diagramas de Estado
- g. Especificidades en la documentación del diseño de sistemas de tiempo real

UNIDAD 12:

Diseño de sistemas web

- a. Conceptos de diseño centrado en el usuario aplicados al diseño de sistemas web
- b. Arquitecturas Web
 - La arquitectura Cliente-Servidor aplicada en el diseño de sistemas web
 - Las tres capas: Presentación, negocio y datos
- c. El modelo Modelo-Vista-Control (MVC).
- d. Patrones aplicables al diseño de aplicaciones web
- e. Diagramas UML de diseño de arquitecturas web
 - i. Diagramas de componentes
 - ii. Diagramas de despliegue
- f. Especificidades en la documentación del diseño de sistemas web

3. BIBLIOGRAFIA

3.1 Básica

- Larman, C., *UML y Patrones*, 2ª. Edicion, Prentice Hall , Madrid 2003, ISBN: 8420534382
- Sommerville I., *Ingeniería del Software*, 7a edición, Pearson Education S.A., Madrid 2005. ISBN: 84-7829-074-5.
- Whitten, J.; Bentley, L. y Dittman, K. *Systems Analysis and Design Methods*. Sixth Edition. McGraw-Hill, , 2004.

3.2 Adicional

Grady Booch, James Rumbaugh, Ivar Jacobson - *El Lenguaje Unificado de Modelado*. Addison Wesley. Madrid, 1999

Jacobson, Rumbaugh, Booch: *El Proceso Unificado*, Pearson Educación S.A., Madrid, 2000

Larman, Craig: *Agile & Iterative Development*, Pearson Educación Inc., Boston, 2003

Peters, J.L.: *Handbook of Software Design*, Yourdon Press, New York, 1988

Pressman, R.S.: *Ingeniería de software - Un enfoque práctico*, 6ª. Edición, Prentice-Hall, Madrid 1997

3.3 Software necesario para desarrollar las clases

1. Microsoft Office

2. Enterprise Architect, Trial edition Enterprise Architect (SPARX Systems):

<http://www.sparxsystems.com/products/ea/trial.html>

User's Guide: <http://www.sparxsystems.com/bin/EAUUserGuide.pdf>

4. METODOLOGIA DE LA ENSEÑANZA

Se realizarán presentaciones conceptuales teóricas sobre cada tema y presentación de casos prácticos de aplicación.

Las clases teóricas serán presentadas casi totalmente usando archivos de PowerPoint proyectados con un cañón conectado a una PC.

Los alumnos, organizados en grupos de 3 a 4 personas, deberán elaborar trabajos prácticos de seguimiento de cada tema, y entregables de un proyecto de diseño de un producto de software en base a lo visto en clase (metas de proyecto). Los entregables que impliquen diseño serán presentados, analizados y las decisiones de diseño defendidas en las clases de trabajos prácticos.

Los trabajos prácticos consistirán en preguntas de concepto relacionadas al material teórico, y en trabajos en los que haya que en modelar sistemas usando la herramienta CASE "Enterprise Architect" y en documentar los modelos usando plantillas especialmente diseñadas.

Además deberán confeccionar una monografía (paper) sobre algún tema planteado en el curso y efectuar una presentación en clase del tema seleccionado.

5. CRITERIOS DE EVALUACION

- 1 Examen parcial, teórico y práctico

- 2 Posibles evaluaciones eventuales sobre temas cerrados (parcialitos)
- 3 Trabajos prácticos requeridos durante el cuatrimestre. Los mismos podrán ser individuales o grupales.
- 4 Examen final teórico con integración de conocimientos prácticos
- 5 Presentismo (conforme las disposiciones vigentes de la UB)

Requisitos de aprobación y promoción de la materia:

- 1 Aprobar el examen parcial 4/10
- 2 Aprobar los trabajos prácticos 4/10
- 3 Aprobar el examen final 4/10
- 4 Requisitos de presencia en las clases (conforme las disposiciones vigentes de la UB).