

FACULTAD DE INGENIERÍA

ÁREA DE COMPUTACIÓN E INFORMÁTICA



Nombre de la materia : Ingeniería de Software A
Clave de la materia: 2301
Clave Facultad:
Clave U.A.S.L.P.: Clave CACEI: CI
Nivel del Plan de Estudios: IC e II:4 No. de créditos: 10
Horas/Clase/Semana: 5 Horas totales/Semestre: 80
Horas/Práctica (y/o Laboratorio): 0
Prácticas complementarias:
Trabajo extra-clase Horas/Semana: 5
Carrera/Tipo de materia: Común del Área / Obligatoria
No. de créditos aprobados:
Fecha última de Revisión Curricular: 25/02/2010
Materia y clave de la materia requisito: Ingeniería de
Requerimientos (2300) (Solo Informática)
Estructura de Datos y Algoritmos B (2222)

PROPÓSITO DEL CURSO

El estudiante analizará y comprenderá las diferentes metodologías, técnicas y procedimientos de arquitectura y diseño de software (SW).

El estudiante comprenderá íntegramente el proceso

unificado de desarrollo en sus fases de diseño así como la notación utilizada en sus artefactos y las herramientas automáticas adecuadas al mismo.

OBJETIVO DEL CURSO

El estudiante aplicará herramientas y técnicas desarrollando arquitecturas y diseños detallados de soluciones de software a partir de especificaciones de

requerimientos de software desarrolladas previamente con técnicas de ingeniería de requerimientos.

El estudiante evaluará diseños desarrollados por terceros probando su calidad y usabilidad.

CONTENIDO TEMÁTICO

1. Fundamentos de Diseño de Software

Tiempo estimado: 6 hrs.

Objetivo: El estudiante conocerá los fundamentos del Diseño en general y de diseño de Software (IS); conocerá las actividades generales del proceso de diseño así como los conceptos involucrados en estas actividades.

1.1 Fundamentos de Diseño

1.1.1 Diseño

1.1.2 Diseño de SW e Ingeniería de SW

1.1.3 Actividades de diseño

1.1.4 Conceptos de Diseño de Software

2 .- Métodos y estrategias de diseño

Tiempo estimado: 10hrs.

Objetivo: El estudiante conocerá y comprenderá las técnicas generales de diseño, su aplicabilidad para diferentes tipos de problemas y las ventajas y desventajas de cada una de ellas.

- 2.1 Diseño orientado a funciones
- 2.2 Diseño centrado en estructuras de datos
- 2.3 Diseño Orientado a Objetos
- 2.4 Diseño Arquitectónico
- 2.5 Diseño de aplicaciones distribuidas
- 2.6 Diseño de aplicaciones en tiempo real
- 2.7 Diseño basado en componentes

3 .- Procesos y notaciones de diseño

Tiempo estimado: 18 hrs.

Objetivo: El estudiante conocerá diferentes metodologías emergentes de diseño de software; comprenderá el lenguaje de modelación UML 2, sus diferentes vistas, diagramas y aplicaciones y aplicará sus conocimientos sobre procesos y notaciones para el desarrollo general de diseños de software.

- 3.1 Diseño en Proceso Unificado de desarrollo
- 3.2 Estado del arte de los procesos de diseño
 - 3.2.1 Metodologías ágiles
 - 3.2.2 Programación extrema

- 3.2.3 Programación Orientada a Aspectos
- 3.3 Lenguaje Unificado de Modelación (UML 2)
 - 3.3.1 Diagramas de clases
 - 3.3.2 Diagramas de objetos
 - 3.3.3 Diagramas de secuencia
 - 3.3.4 Diagramas de colaboración
 - 3.3.5 Diagramas de estados
 - 3.3.6 Diagramas de casos de uso
 - 3.3.7 Diagramas de despliegue
 - 3.3.8 Diagramas de actividad
 - 3.3.9 Diagramas de componentes

4 .-Herramientas de Diseño

Tiempo estimado: 6 hrs.

Objetivo: El estudiante conocerá y aplicara herramientas automatizadas y herramientas CASE para el diseño de software.

- 4.1 Herramientas estándar
- 4.2 Herramientas para el proceso Unificado de desarrollo y el lenguaje UML
 - 4.2.1 Rational Rose

5 .-Diseño de Software

Tiempo estimado: 14 hrs.

Objetivo: El estudiante comprenderá el proceso unificado de desarrollo con énfasis en las etapas de diseño.

El estudiante comprenderá los conceptos de diseño de software.

El estudiante analizará diferentes problemas y evaluará diferentes estrategias de solución

Siguiendo las etapas del proceso unificado de desarrollo, partir de especificaciones de requerimientos de software (desarrolladas con técnicas de ingeniería de requerimientos), diseñará soluciones a nivel de diseño detallado.

- 5.1 Proceso de Diseño
- 5.2 Conceptos de Diseño
 - 5.2.1 Abstracción
 - 5.2.2 Modularidad
 - 5.2.3 Ocultamiento de la información
 - 5.2.4 Concurrencia
 - 5.2.5 Control y manejo de eventos
 - 5.2.6 Manejo de excepciones
- 5.3 Prácticas de Diseño de Software
 - 5.3.1 Etapas de diseño en el RUP
 - 5.3.2 Casos prácticos

6 .-Calidad en el diseño de Software

Tiempo estimado: 6 hrs.

Objetivo: El estudiante conocerá los principios atributos y modelos de calidad para el software

El estudiante comprenderá la importancia del aseguramiento de la calidad en el proceso así como el producto del diseño de software.
El estudiante aplicará técnicas de evaluación de la calidad de diseños de software.

- 6.1 Antecedentes
- 6.2 Atributos
- 6.3 Análisis de Calidad
- 6.4 Modelos de calidad
 - 6.4.1 Modelo de la organización internacional de estándares (ISO)
 - 6.4.2 Modelo de capacidad y madurez (CMM) de SEI (Software Eng. Institute)
- 6.5 Técnicas de evaluación

7 .-Arquitectura de Software

Tiempo estimado: 10 hrs.

Objetivo: El estudiante conocerá los elementos esenciales de la arquitectura de software y entenderá la importancia de la misma.

El estudiante conocerá los diferentes modelos y estilos arquitectónicos de diseño de software.

Aplicará de manera general los patrones de diseño al desarrollo arquitectónico de Software

El estudiante aplicará diferentes arquitecturas en el desarrollo del diseño general de aplicaciones

- 7.1 Fundamentos de arquitectura de Software
- 7.2 Estilos arquitectónicos
- 7.3 Modelos y marcos de trabajo
- 7.4 Patrones de diseño arquitectónico
- 7.5 Arquitecturas alternativas
 - 7.5.1 Flujos de datos
 - 7.5.2 Componentes
 - 7.5.3 Máquinas virtuales
 - 7.5.4 Arquitecturas de almacenamiento
 - 7.5.5 Arquitecturas de capas

8 .-Interacción Humano – Computadora

Tiempo estimado: 10 hrs.

Objetivo: El estudiante conocerá los aspectos involucrados en la interacción humano – computadora (IHC) y su importancia

El estudiante conocerá los modelos de estudio y desarrollo de la IHC así como los límites en el ámbito de esta disciplina.

El estudiante comprenderá el impacto de un mal y de un buen diseño de interfaces; aplicará técnicas de evaluación de la calidad y usabilidad de diseños de software.

- 8.1 Naturaleza de la Interacción Humano – Computadora (IHC)
 - 8.1.1 Metamodelos
- 8.2 Uso y contexto de las computadoras
- 8.3 Características del humano

8.3.1	Procesamiento de información	8.5.1	Usabilidad
8.3.2	Lenguaje, comunicación e interacción	8.5.2	Técnicas de aseguramiento de la calidad de IHC
8.3.3	Ergonomía		
8.4	Interfaces		
8.4.1	Técnicas de diálogo		
8.4.2	Gráficas		
8.4.3	Diseño		
8.5	Calidad en IHC		

METODOLOGÍA

Exposición de temas por parte del profesor, uso de cañón proyector y de proyector de acetatos, análisis de

conceptos teóricos, desarrollo de temas utilizando herramientas de apoyo, trabajo grupal e individual.

EVALUACIÓN

De acuerdo con el reglamento de exámenes, la calificación final se obtiene del promedio de los cinco exámenes parciales; para tener derecho a cada examen

parcial se debe entregar avance del proyecto y para reportar la calificación final, el proyecto terminado.

BIBLIOGRAFÍA

Bibliografía Básica

Ingeniería de software
 Roger S. Pressman
 Mc Graw Hill
 Aravaca (Madrid) 2002

Ingeniería de Software
 Ian Sommerville
 Addison Wesley
 Madrid (España) 2005

Proceso Unificado de Desarrollo de Software
 James Raumbaugh, Ivar Jacobson y Grady Booch
 Addison Wesley

Bibliografía Complementaria

Ingeniería de software orientada a objetos con UML
 Alfredo Weitzenfeld
 Thompson
 México, D.F. 2004

Design Patterns
 Erich Gamma, Richard Helm, Ralph Jonson, John Vlissides
 Addison Wesley