

Nombre de la asignatura: **MODELADO ORIENTADO A OBJETOS**

Línea de Trabajo: **Ingeniería de Software**

Tiempo de dedicación del estudiante a las actividades de:

DOC	TIS	TPS	Horas Totales	Créditos
48	20	100	168	6

DOC: Docencia; **TIS:** Trabajo independiente significativo; **TPS:** Trabajo profesional supervisado

1. Historial de la Asignatura.

Lugar y Fecha de Elaboración o Revisión	Participantes	Observaciones (cambios y justificaciones)
21 nov 2011	MC Osvaldo Daniel Fernández Bonilla	

2. Pre-requisitos

Ninguno

3. Objetivo de la Asignatura

Aplicar la práctica de Modelado de arquitecturas orientadas a objetos con UML, así como los principios de diseño para el desarrollo eficiente de productos de software de alta calidad.

4. Aportación del Perfil del Graduado

Esta materia contribuirá a la formación de una capacidad analítica y responsable con respecto al análisis y diseño de sistemas de alta calidad.

La necesidad actual de desarrollar sistemas con las más modernas técnicas de diseño es hoy por hoy un gran reto. El estudiante aprenderá, aplicará y será capaz de innovar en cuanto al mejor desarrollo de sistemas, dejando atrás el desarrollo de software empírico

5. Contenido Temático

Unidad	Temas	Subtemas
1	Introducción al modelado orientado a objetos	1.1. La evolución del modelado de objetos 1.2. Fundación del modelado de objetos 1.3. Elementos del modelado de objetos 1.4. Diagramas UML 1.5. Diagramas de casos de uso 1.6. Clases y objetos 1.7. Diagramas de clases 1.8. Diagramas de interacción 1.9. Diagramas de estado maquina 1.10. Paquetes 1.11. Comunicación entre diagramas 1.12. Diagramas de aplicación
2	Diseño del software	2.1. Introducción. 2.1.1.Importancia. 2.1.2.Definiciones. 2.1.3.Evolución del diseño. 2.2. Procesos de diseño de software. 2.2.1.Diseño de datos. 2.2.2.Diseño arquitectónico. 2.2.3.Diseño a nivel de componentes. 2.2.4.Diseño de la interfaz. 2.3. Principios fundamentales del diseño. 2.3.1.Abstracción. 2.3.2.Refinoamiento sucesivo. 2.3.3.Ocultación de la información. 2.3.4.Modularidad. 2.3.5.Arquitectura de software. 2.3.6.Jerarquía de control (estructura del Programa). 2.4. Estructura de los datos.
3	Ciclos y vida del software	3.1. Modelos de ciclos de vida del software 3.2. Designación, verificación y validación 3.3. Actividades y pruebas de un ciclo de vida de software 3.4. Modelo codifica y arregla 3.5. Modelo de cascada 3.6. Modelo de código abierto 3.7. Procesamiento con el estándar agil 3.8. Modelo sincronizado estable 3.9. Modelo en espiral 3.10. Comparativa de los diferentes modelos

4	Patrones de diseño	<ul style="list-style-type: none"> 4.1. Introducción a los patrones de diseño 4.2. Selección y uso de un diseño de patrones 4.3. Creacionales 4.4. Método de fabrica 4.5. Estructurales 4.6. Método de adaptador 4.7. De conducta 4.8. Método de plantillas
5	Medidas de calidad del software	<ul style="list-style-type: none"> 5.1. Métricas estáticas. Mediciones hechas en las representaciones del sistema como diseño. 5.2. Métricas dinámicas. Mediciones hechas en el programa en ejecución. 5.3. Atributos de software de calidad
6	Lenguajes orientados a objetos	<ul style="list-style-type: none"> 6.1. Riegos y beneficios del uso de un lenguaje orientado a objeto 6.2. Lenguajes orientados a objetos existentes 6.3. Aplicaciones reales

6. Metodología de Desarrollo del Curso

Queda a elección del docente manejar un problema específico para cada una de las unidades de la 3 a la 6, o bien un solo proyecto integrador de todos los patrones estudiados

7. Sugerencias de Evaluación

Diseño de un Modelo en UML para la unidad 1 y 2.

Elaboración de programas hechos en lenguajes orientados a objetos sobre la aplicación de cada uno de los patrones estudiados en las unidades 2, 3 y 4.

Exámenes parciales escritos al final de cada unidad.

Un proyecto de desarrollo de un sistema donde se apliquen los conocimientos adquiridos de diseño de patrones.

8. Bibliografía y Software de Apoyo

Object-Oriented Analysis and Design with Applications Third Edition Grady Booch Robert A. Maksimchuk Michael W. Engle Bobbi J. Young, Ph.D. Jim Conallen Kelli A. Houston

SOFTWARE MODELING AND DESIGN UML, Use Cases, Patterns, and Software Architectures Hassan Gomaa George Mason University, Fairfax, Virginia

Design Patterns Elements of Reusable Object-Oriented Software Erich Gamma Richard Helm Ralph Johnson John Vlissides

DESIGN PATTERNS IN JAVA™ Steven John Metsker William C. Wake

Growing Object-Oriented Software, Guided by Tests Steve Freeman and Nat Pryce

Object-Oriented Programming and Java Second edition Danny Poo Derek Kiong Swarnalatha Ashok

Use Case Driven Object Modeling with UML Theory and Practice Doug Rosenberg and Matt Stephens

Head First Object-Oriented Analysis and Design Brett D. McLaughlin Gary Pollice David West

Object-Oriented Software Engineering Stephen R. Schach

9. Actividades Propuestas

Se sugiere que las prácticas propuestas sean realizadas por equipos para estar en concordancia con la finalidad de fomentar la discusión de ideas que plantea el curso. En este sentido, se proponen las siguientes prácticas por unidad:

Unidad	Actividad
1	Desarrollar un modelo UML acorde a un proyecto que se le estipule
2	Desarrollar ensayos y tablas comparativas sobre los diferentes tipos de arquitecturas de software que existen.
3	Crear una comparativa de los ciclos de vida del software y hacer un escrito sobre ellos

4	Exponer diferencias entre cada uno de los patrones y sus tipos y porque debe aplicarse alguno en particular a un sistema
5	Analizar y medir un software que se desarrolle tanto propio como de otros compañeros del grupo
6	Comparar los diferentes lenguajes que existen y las ventajas y desventajas de ellos.

10. Nombre y Firma del Catedrático Responsable



M.C en C.C Osvaldo Daniel Fernández Bonilla