



INFORMACIÓN GENERAL

Materia	Principios de Diseño de Software
Titular / Cotitular	Alejandro García Fernández / Perla Velasco Elizondo
Fecha de elaboración	7-junio-2015

INTRODUCCIÓN GENERAL DE LA MATERIA

El diseño es una de las habilidades básicas del desarrollo de sistemas de software. Por ello, antes de diseñar sistemas completos, es necesario saber cómo diseñar programas.

Este curso busca dotar al alumno de los conceptos, técnicas y herramientas que le permitan desarrollar la habilidad de diseñar programas que serán implementados en un lenguaje de programación orientado a objetos.

Con esto se pretende elevar la calidad de los programas que el alumno desarrolla, lo que será indispensable en las siguientes materias de la maestría y en su vida laboral.

OBJETIVO GENERAL

Al término de este curso el alumno será capaz de diseñar programas que serán implementados con calidad en un lenguaje de programación orientado a objetos.

OBJETIVOS PARTICULARES

Al finalizar este curso el alumno será capaz de:

1. Identificar y comprender los conceptos fundamentales del diseño de sistemas, especialmente los del paradigma orientado a objetos y del diseño basado en el dominio.
2. Identificar y comprender el concepto de patrón de diseño, especialmente los relacionados al diseño de sistemas orientados a objetos.
3. Identificar el lenguaje de diseño UML así como comprender la sintaxis y semántica de algunos de sus diagramas y su mapeo a código fuente.
4. Aplicar estos conceptos así como técnicas y herramientas relacionadas en el diseño e implementación de un problema del mundo real.

TEMARIO

1. Diseño de Sistemas de Software
 - 1.1. Definición y Objetivos
 - 1.2. Principales Artefactos.
 - 1.2.1. Modelo estático
 - 1.2.2. Modelo dinámico
 - 1.2.3. Modelo de datos
 - 1.3. Principios de Diseño.
 - 1.3.1. Modularidad
 - 1.3.2. Cohesión
 - 1.3.3. Acoplamiento
2. Código de Calidad (Clean Code)
 - 2.1. Fundamentos
 - 2.2. Principios SOLID
 - 2.3. Diseño de Componentes
3. Análisis y Diseño Orientado a Objetos
 - 3.1. Qué es la Paradigma Orientado a Objetos
 - 3.2. Cómo es diferente Paradigma Orientado a Objetos de otros Paradigmas
 - 3.3. Diseño Guiado por el Dominio (Domain Driven Design)
4. Lenguaje de Modelado
 - 4.1. Diagramas Estáticos
 - 4.2. Diagramas Dinámicos
5. Patrones de Diseño (Design Patterns)
 - 5.1. Patrones creacionales
 - 5.2. Patrones estructurales
 - 5.3. Patrones de comportamiento
6. Rediseño con patrones (Refactoring to Patterns)

*Se recomienda que los diagramas de UML sean introducidos, poco a poco al ir cubriendo los demás temas del plan de estudio.

BIBLIOGRAFÍA

No.	Título	Autor	Editorial	Año
1	Clean Code: A Handbook of Agile Software Craftsmanship	Robert C. Martin	Prentice Hall	2008
2	UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)	Martin Fowler	Addison-Wesley Professional	2003
3	Domain Driven Design Quickly	Abel Avram	infoq	2006



4	Head First Design Patterns	Eric Freeman, Bert, Bates, Sierra, Robson	O'Reilly Media	2004
5	Design Patterns: Elements of Reusable Object-Oriented Software	Gamma, Helm, Johnson, Vlissides	Addison-Wesley Professional	1994
6	Holub on Patterns Learning Design Patterns by Looking at Code	Allen Holub	Apress	2004
7	Refactoring to Patterns	Joshua Kerievsky	Addison-Wesley Professional	2004

EVALUACIÓN

ASPECTO A EVALUAR	PORCENTAJE
Proyecto de code review usando clean code	15%
Proyecto de implementación de patrones	15%
Proyecto de patrones aplicados a refactoring	15%
Tareas y Exámenes rápidos	55%