

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO****Licenciatura en Ciencias de la Computación****Facultad de Ciencias**

Programa de la asignatura

**Denominación de la asignatura:*****Patrones de Diseño de Software***

Clave:	Semestre: 7-8	Eje temático: Ingeniería de Software			No. Créditos: 10
Carácter: Optativa		Horas		Horas por semana	Total de Horas
Tipo: Teórico-Práctica		Teoría:	Práctica:	7	112
		3	4		
Modalidad: Curso		Duración del programa: Semestral			

Asignatura con seriación indicativa antecedente: Ingeniería de Software**Asignatura con seriación indicativa subsecuente:** Ninguna**Objetivos generales:**

Identificar y resolver problemas que se suelen presentar los patrones de diseño.

Escoger el patrón de diseño adecuado.

Implementar la solución utilizando el patrón adecuado.

Construir programas bien estructurados, fáciles de entender y corregir.

Utilizar correctamente los patrones de diseño.

Índice temático

Unidad	Temas	Horas	
		Teóricas	Prácticas
I	Introducción a UML y al diseño orientado a objetos	6	8
II	Análisis y diseño de patrones	6	8
III	Catálogo de patrones	21	28
IV	Aplicación práctica de los patrones de diseño	12	16
V	Análisis y discusión sobre la necesidad de los patrones de diseño	3	4
Total de horas:		48	64
Suma total de horas:		112	

Contenido temático	
Unidad	Tema
I Introducción a UML y al diseño orientado a objetos	
I.1	UML. Diagramas necesarios para describir los patrones de diseño, su interacción y comportamiento.
I.2	Problemas asociados con el diseño orientado a objetos.
I.3	Granularidad, interfaces de los objetos, implementaciones.
I.4	Importancia de reutilizar.
I.5	Evolución del software.
II Análisis y diseño de patrones	
II.1	Concepto de patrón de diseño.
II.2	Características que conforman un patrón de diseño: nombre, problema que resuelve y situaciones habituales en donde se presenta, solución al problema y consecuencias.
II.3	Aplicación de patrones en entornos de desarrollo de software.
III Catálogo de patrones	
III.1	Patrones de creación: fábrica abstracta, constructor, método de fábrica, inicialización floja, prototipo y solitario.
III.2	Patrones de estructura: adaptador, puente, composición, decorador, fachada, peso ligero y apoderado (Proxy).
III.3	Patrones de comportamiento: cadena de responsabilidad, ejecutor, intérprete, iterador, mediador, recuerdo, observador, estado, estrategia, método de plantilla y visitante.
III.4	Patrones arquitectónicos. MVC.
IV Aplicación práctica de los patrones de diseño	
IV.1	Aspectos concretos del uso de patrones en lenguajes de programación orientados a objetos.
IV.2	Aplicación inmediata de los patrones a problemas concretos.
V Análisis y discusión sobre la necesidad de los patrones de diseño	

Bibliografía básica:

1. Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1994.
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, *Pattern-Oriented Software Architecture. A System of Patterns*, John Wiley and Sons, 1996.

Bibliografía complementaria:

1. Alan Shalloway, James R. Trot., *Design Patterns Explained: A New Perspective on Object-Oriented Design* (2nd Edition) Addison-Wesley Professional. 2004.
2. John Vlissides, *Pattern Hatching: Design Patterns Applied*, Addison-Wesley Professional. 1998.
3. Steve McConnell, *Code Complete: A Practical Handbook of Software Constructions*, Microsoft Press: 2nd edition, 2004.

4. Martin Fowler, *UML Distilled: A Brief Guide to the Standard Object Oriented Modeling Language*, 3rd Edition. 2003.
5. Peter Coad and Edward Yourdon, *Object-Oriented Design*, Prentice Hall, 1991.

Sugerencias didácticas:		Métodos de evaluación:	
Exposición oral	(X)	Exámenes parciales	(X)
Exposición audiovisual	(X)	Examen final escrito	(X)
Ejercicios dentro de clase	(X)	Trabajos y tareas fuera del aula	(X)
Ejercicios fuera del aula	(X)	Prácticas de laboratorio	()
Seminarios	()	Exposición de seminarios por los alumnos	()
Lecturas obligatorias	(X)	Participación en clase	()
Trabajo de investigación	()	Asistencia	()
Prácticas de taller o laboratorio	(X)	Proyectos de programación	()
Prácticas de campo	()	Proyecto final	()
		Seminario	()
Otras: _____		Otras: _____	

Perfil profesiográfico:

Egresado preferentemente de la Licenciatura en Ciencias de la Computación o matemático con especialidad en computación con amplia experiencia de programación. Es conveniente que posea un posgrado en la disciplina. Con experiencia docente.