

PROGRAMA DE ESTUDIOS: DISEÑO DE SOFTWARE

PROTOCOLO

Fechas	Mes/año
Elaboración	05-2010
Aprobación	
Aplicación	09-2010

Clave			Semestre	7°	
Nivel	Licenciatura	X	Maestría		Doctorado
Ciclo	Integración		Básico		Superior
Colegio	H. y C.S.		C. y T.	X	C. y H.

Plan de estudios del que forma parte: Ingeniería de Software

Propósito(s) general(es):
El estudiante deberá ser capaz de aplicar los fundamentos, métodos, técnicas y herramientas de diseño de software.

Carácter		Modalidad			Horas de estudio semestral (16 semanas)						
Indispensable	X	Seminario		Taller		Con Docente	Teóricas	36	Autónomas	Teóricas	20
		Curso	X	Curso-taller			Prácticas	36		Prácticas	36
Optativa *		Laboratorio	X	Clínica		Carga horaria semanal: 4.5 + 3.5 = 8		Carga horaria semestral:		128	

Asignaturas Previas:	Asignaturas Posteriores:
Análisis de algoritmos, Análisis y modelamiento de software. Lenguajes de programación	Arquitectura de software

Requerimientos para cursar la asignatura	Contenido
	Conocimientos: dominio de conceptos fundamentales de programación: objetos, herencia, clases, variables, estructuras de control, funciones y arreglos. Habilidades: para la búsqueda sistemática de soluciones a través del modelado de componentes de software, capacidad para plantear y resolver problemas por medio del razonamiento lógico.

Perfil deseable del profesor: Con Maestría en Ingeniería de Software o algún grado de maestría afín, o bien, amplios conocimientos y experiencia laboral en el área. Además, amplia experiencia docente y de investigación.

Academia responsable del programa:	Diseñador (es):
Informática	Profr. Jorge Wals Selvas

* Aquellas en las que se ofrece la posibilidad de cursar una de las asignaturas, para cubrir un requisito INDISPENSABLE será considerada INDISPENSABLE.

INTRODUCCIÓN

La función de la materia Diseño de Software dentro de la Licenciatura en Ingeniería de software es la de proporcionar al estudiante métodos y técnicas que sirvan para desarrollar software de calidad. Para llevar a cabo este fin se modelaran sistemas de software utilizando como herramienta principal al UML, el Lenguaje Unificado de Modelado, que es hoy en día la forma más utilizada y práctica en las organizaciones para desarrollar software durable y de calidad.

En esta materia al estudiante adquirirá los conocimientos que le permitirán modelar y diseñar sistemas de software, que puedan ser usados en la industria o escuela.

Para su estudio se deben tener conocimientos de Análisis y modelamiento de software (1-CT-IS-13).

PROPÓSITOS GENERALES

El estudiante deberá ser capaz de aplicar los fundamentos, métodos, técnicas y herramientas de diseño de software.

CONTENIDOS

TEMAS Y SUBTEMAS	PROPÓSITOS ESPECIFICOS
1 La etapa de diseño en el Proceso Unificado 1.1 Principios del modelado 1.2 Modelado Orientado a Objetos 1.3 Presentación de UML y notación	El estudiante comprenderá la importancia de modelar un sistema, los principios básicos del modelado y adquirirá los conceptos preliminares del modelado Orientado a Objetos con la notación de UML.
2 Métodos, herramientas y técnicas para la especificación del diseño con UML 2.1 Modelado básico: clases, relaciones, diagramas de clases 2.2 Modelado avanzado: características avanzadas de las clases y relaciones, interfaces 2.3 Paquetes, instancias y diagramas de objetos	El estudiante identificará los principales diagramas de UML, para aplicarlos a los diseños de software. Utilizando los paradigmas de la Orientación a Objetos.
3 El proceso de administración de la etapa de diseño 3.1 Modelado básico del comportamiento 3.2 Diagramas de caso de uso, de interacción y de actividades	El estudiante analizará la relación entre los componentes de un sistema, a partir de tres aspectos fundamentales de UML, el comportamiento, los casos de uso y los diagramas de actividades, para diseñar de manera eficiente la administración de los recursos de un sistema.

4 Aseguramiento de la calidad en la etapa del diseño	El estudiante aplicará las técnicas y diagramas de UML para garantizar que el diseño de software sea adecuado a la necesidad de usuarios que le dio origen.
4.1 Modelado avanzado del comportamiento: eventos y señales, maquinas de estados	
4.2 Procesos e hilos	
4.3 Diagramas de estado	

METODOLOGÍA PARA EL CURSO

Se trata de un curso teórico práctico, que se impartirá por medio de tres elementos: clases teóricas, prácticas y asesorías, de acuerdo al modelo educativo de la UACM. Estos tres elementos se efectuarán en cualquiera de las siguientes modalidades: conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, lluvia de ideas, uso de software, grupos de discusión, trabajo en equipo, análisis y demostración de problemas, diseño de un prototipo, investigación bibliográfica.

EVALUACIÓN DIAGNÓSTICA

Se aplicará un examen escrito para verificar si el estudiante cumple con los conocimientos previos requeridos, los conceptos previos a evaluar serán principalmente los adquiridos en el curso de Análisis y modelamiento de software.

EVALUACIÓN FORMATIVA

Se recomienda realizar al menos una evaluación formativa al finalizar cada una de las unidades temáticas para que el estudiante tenga una retroalimentación adecuada sobre la adquisición de los conocimientos durante el semestre y no hasta el final del mismo, de esta forma se cuenta con tiempo para que el estudiante realice ajustes sobre su desempeño académico en el curso. De tal suerte que en total se realizaran cuatro evaluaciones formativas:, una por unidad temática, de acuerdo a la siguiente lista:

- La etapa de diseño en el Proceso Unificado.
- Métodos, herramientas y técnicas para la especificación del diseño con UML.
- El proceso de administración de la etapa de diseño.
- Aseguramiento de la calidad en la etapa del diseño.

EVALUACIÓN DE CERTIFICACIÓN

La evaluación de certificación, constara de dos partes que se explican a continuación:

La parte I es el instrumento o evaluación *conceptual-práctica*, tiene una ponderación del 60% y se puede aplicar a lo largo del semestre con prácticas de laboratorio definidas por cada Profesor, o bien, presentar una evaluación escrita final definida por el comité de certificación, que consta de preguntas y ejercicios de acuerdo con el contenido temático.

La parte II que es el restante 40% de la evaluación consiste en un *proyecto* de programación en donde el estudiante aplicará los conocimientos adquiridos. El estudiante deberá entregar a su profesor un documento impreso con el análisis, modelado UML y la implementación de su diseño final, además de una copia electrónica del mismo.

BIBLIOGRAFÍA

La bibliografía que se enlista a continuación se considera como básica para estudiante y profesor:

Núm.	Bibliografía	Temas para los que se recomienda
1	Booch, G., Diseño Orientado a Objetos con Aplicaciones. Addison Wesley. 1996.	1, 2, 3, 4
2	Gamma, E. Design Patterns. Addison Wesley. 1995.	1, 2, 3, 4
3	Kendall y Kendall. Análisis y Diseño de Sistemas. 3a. Ed. Prentice Hall, 1997.	1, 2, 3, 4
4	Rumbaugh, J., Blaha, M., Prenerlani, W. Object Oriented modeling. Prentice Hall Press. 1991.	1, 2, 3, 4

OTROS RECURSOS

El laboratorio de cómputo, deberá contar con los siguientes recursos: 30 computadoras personales con procesador a 3GH conectadas en red y con internet, dispositivo óptico cdrw/dvd 24x, 2 Gb de RAM, 160 GB en disco, monitor a color de alta resolución. 1 Impresora laser. 1 Servidor para ambiente de redes, procesador xeon 2.8ghz/1mb con bus de 800 mhz, 2Gb de RAM, disco duro SCSI de 160 gb, dispositivo óptico cdrw/dvd 24x, adaptador de red (necesario a partir del tercer año de la carrera).