UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA
Clave: 08MSU0017H

FACULTAD INGENIERÍA
Clave:

PROGRAMA DEL CURSO:

Diseño de Software a Bajo Nivel

DES:	INGENIERÍA
Programa(s) Educativo(s):	Ingeniería de Software
Tipo de materia:	Obligatoria
Clave de la materia:	IS604
Semestre:	Sexto
Área en plan de estudios:	Ciencias de la Computación e Informática
Créditos	4
Total de horas por semana:	4
Teoría:	3
Práctica	
Taller:	
Laboratorio:	1
Prácticas complementarias:	
Trabajo extra clase:	
Total de horas semestre:	64
Fecha de actualización:	09/06/2011
Clave y Materia requisito:	**Ninguna**

Propósitos del Curso:
El alumno aprende a utilizar habilidades que ha desarrollado a lo largo de las asignaturas cursadas hasta ahora facilitar algunos de los aspectos de “bajo nivel” que tienen que ver con la implementación de un proyecto de software. En particular, el alumno aprende como aprovechar el uso de estructuras y patrones de diseño en lo referente a la implementación de un sistema. Particular énfasis se pone en el uso de patrones de diseño, especialmente en un contexto de orientación a objetos. También se cubre en detalle el uso de la ingeniería en reversa como metodología para rediseñar un sistema, o para facilitar su mejora y mantenimiento

Al final del curso el estudiante:

- Implementa el diseño de un sistema en primera instancia y el rediseño y mantenimiento de un sistema heredado
- Implementa herramientas y modelos para la manipulación de datos y estructuración de sistemas de software basándose en patrones de diseño.
- Analiza un sistema donde puede hacer el reuso de una sistema previamente implementado
- Aplica la ingeniería en reversa para la recuperación de un software ya hecho.

COMPETENCIAS (Tipo y Nombre de las Competencias que nutren a la materia y a las que contribuye)	**CONTENIDOS** (Unidades, Temas y Subtemas)	**RESULTADOS DE APRENDIZAJE** (Por Unidad)
El curso promueve las siguientes competencias:	**UNIDAD I: DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE SOFTWARE EN DETALLE**	Expresa una visión o perspectiva muy general de lo que implica el diseño de un sistema de

Ciencias de la Computación e Informática. Uso de Información. Solución de Problemas. Trabajo en equipo**.**	1.1.- Métodos de diseño. 1.2.- Fundamentos de patrones de diseño. 1.3.- Diseño de componentes. 1.4.- Diseño de componentes e interfaces del sistema.	software. Es posible que este tema sea estudiado concurrentemente en otras materias, así que aquí solo se pretende resumir los conceptos de diseño de forma general.
	UNIDAD II: REDISEÑO Y PATRONES DE DISEÑO 2.1.- Patrones basados en un enfoque algorítmico. 2.2.- Patrones basados en diseño computacional. 2.3.- Patrones basados en modelos de ejecución. 2.4.- Patrones basados en estrategias de implementación. 2.5.- Patrones basados en diseño estructural. 2.6.- Rediseño del software. 2.6.1.- Rediseño de un software heredado. 2.6.2.- Rediseño de un software anterior. 2.6.3.- Reversionando un software	Describe el concepto de un patrón de diseño, y comprende los diferentes criterios en que un patrón de diseño se puede utilizar. Analiza cuando es propio el uso de uno o de otro, y los efectos que el utilizarlos tiene sobre el sistema de software final. Identifica también las implicaciones que el rediseño de un proyecto de software implica, así como los alcances.
	UNIDAD III: INTRODUCCIÓN A LOS ENFOQUES FORMALES PARA DISEÑO 3.1.- Definición formal inicial del proyecto de software. 3.2.- Desarrollo formal y verificación formal del proyecto de software. 3.3.- Verificación formal del proyecto de software.	Indica como el estudio de las técnicas matemáticas formales para el planteamiento de un argumento y su prueba, son aplicados también en la especificación, implementación y validación de un proyecto de software.
	UNIDAD IV: ANÁLISIS DEL DISEÑO BASADO EN CRITERIOS DE CALIDAD 4.1.- Confiabilidad en el diseño del software. 4.1.1.- Requerimientos. 4.1.2.- Diseño. 4.1.3.- Programación. 4.1.4.- Construcción y ejecución. 4.1.5.- Prueba. 4.1.6.- Ejecución. 4.2.- Factores de calidad en software. 4.2.1.- Legibilidad 4.2.2.- Completitud y consistencia.	Reconoce el concepto de calidad como un criterio presente siempre en todas las fases de la implementación de un proyecto de software. Demuestra cómo afecta cada etapa del proceso, y comprende también bajo qué criterios estas etapas se verán afectadas.

	4.2.3.- Portabilidad y consistencia. 4.2.4.- Mantenibilidad y verificabilidad. 4.2.5.- Usabilidad y confiabilidad. 4.2.6.- Eficiencia y seguridad.	
	UNIDAD V: MEJORA DEL DESEMPEÑO Y DE LA MANUTENCIÓN DEL SOFTWARE 5.1.- Ingeniería del desempeño. 5.1.1.- Incepción del proyecto de software. 5.1.2.- Elaboración del proyecto de software. 5.1.3.- Construcción del proyecto de software. 5.1.4.- Transición del proyecto de software. 5.2.- Principios de manutención del proyecto de software. 5.2.1.- El diseño consiente. 5.2.2.- El de “la cosa esplendorosa”. 5.2.3.- El de los requerimientos multi-nivel. 5.2.4.- El de nivel de paga. 5.2.5.- El de la prioridad de dinámica.	Emplea el diseño de un proyecto de software apegado a los criterios de ingeniería de desempeño, comprende como el desempeño esperable de un software limita o afecta las decisiones de diseño, y comprende también que la manutención de su proyecto vista desde distintos criterios es también un factor a tener en cuenta.
	UNIDAD VI: INGENIERÍA EN REVERSA EN SOFTWARE. 6.1.- Recolección de datos e información. 6.2.- Análisis de los datos recolectados. 6.3.- Extracción de estructuras y patrones. 6.4.- Determinación de funcionalidad en cada estructura. 6.5.- Determinación del flujo de datos entre estructuras. 6.6.- Determinación del flujo de control entre estructuras. 6.7.- Revisión y validación del modelo recolectado.	Demuestra el concepto de ingeniería en reversa como una opción más para el proceso de implementación de un proyecto de software. El alumno clasifica las etapas del proceso de ingeniería en reversa, y aprende a llevarlo a cabo con los resultados medibles que se esperan en cada etapa.
	UNIDAD VII: ENFOQUES AL CAMBIO DE DISEÑO 7.1.- Razones para cambiar el diseño. 7.2.- Niveles de cambio del diseño. 7.2.1.- Solo en módulos. 7.2.2.- En el sistema completo.	Desarrolla las opciones que tiene para cambiar el diseño de un proyecto de software. En caso de ser necesario, el alumno anticipa en que etapas el cambio es necesario, y conoce las implicaciones

	7.3.- Impacto del cambio en el sistema. 7.4.- Cambio en diseño y herramientas de diseño.	que sobre distintos criterios, este cambió provoca, y sabe sortearlas de la mejor manera.

OBJETO DE ESTUDIO	**METODOLOGIA** (Estrategias, secuencias, recursos didácticos)	**EVIDENCIAS DE APRENDIZAJE**
UNIDAD I: DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE SOFTWARE EN DETALLE UNIDAD II: REDISEÑO Y PATRONES DE DISEÑO. UNIDAD III: INTRODUCCIÓN A LOS ENFOQUES FORMALES PARA DISEÑO. UNIDAD IV: ANÁLISIS DEL DISEÑO BASADO EN CRITERIOS DE CALIDAD UNIDAD V: MEJORA DEL DESEMPEÑO Y DE LA MANUTENCIÓN DEL SOFTWARE UNIDAD VI: INGENIERÍA EN REVERSA EN SOFTWARE UNIDAD VII: ENFOQUES AL CAMBIO DE DISEÑO	Lectura. Lectura Comentada Expositiva Materiales Gráficos: artículos, libros, Cañón Pizarrón	Se recomienda que le catedrático que imparta ésta materia tenga contacto con las normas que rigen al software. Tareas de Investigación Prácticas de Laboratorio Exposiciones

FUENTES DE INFORMACIÓN (Bibliografía/Lecturas por unidad)	**EVALUACIÓN DE LOS APRENDIZAJES** (Criterios e instrumentos)
1. B. B. Agarwal, M. Gupta, S. P. Tayal. (2009) *Jones & Bartlett Learning.* 2. Sommerville Ian. (2005). *Ingeniería del Software*. (6a Ed.) Pearson 3. Dustin Elfriede, Rashka Jeff, Pau John. (1999). *Automated Software Testing: Introduction,*	Se toma en cuenta para integrar **calificaciones parciales:** 3 exámenes parciales escritos donde se evalúa conocimientos, comprensión y aplicación. Con un valor del 30%, 30% y 40% respectivamente Nota: para acreditar el curso se deberá tener calificación aprobatoria tanto en la teoría como en las prácticas. La calificación mínima aprobatoria

Management and Performance. Addison-Wesley.	será de 6.0

Cronograma del Avance Programático

S e m a n a s

Unidades de aprendizaje	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I: DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE SOFTWARE EN DETALLE.																
II: REDISEÑO Y PATRONES DE DISEÑO.																
III:- INTRODUCCIÓN A LOS ENFOQUES FORMALES PARA DISEÑO.																
IV: CONFIABILIDAD EN EL DISEÑO DEL SOFTWARE.																
V: MEJORA DEL DESEMPEÑO Y DE LA MANUTENCIÓN DEL SOFTWARE.																
VI: INGENIERÍA EN REVERSA EN SOFTWARE.																
VII: ENFOQUES AL CAMBIO DE DISEÑO.																