



Nombre de la asignatura								Ingeniería de Software	Clave de la asignatura	
Área de formación	Docencia frente a grupo según SATCA				Trabajo de Campo Supervisado según SATCA				Carácter de la asignatura	
	HCS	HPS	TH	C	HTCS	TH	C	TC	(X) Obligatoria	() Optativa
Sustantiva Profesional	2	2	4	4	0	0	0	4		

SERIACIÓN		
Explícita		Implícita
Asignaturas antecedentes		Conocimientos previos
Ninguna	Laboratorio de Diseño de Software	Manejo de lenguajes de programación estructurada y orientado a objetos.

PROPOSITO DE LA ASIGNATURA	
Aplicar modelos, técnicas y herramientas para los requisitos y diseño del ciclo de vida de desarrollo de software.	
COMPETENCIAS A DESARROLLAR	
Genéricas	Específicas
Capacidad de organizar y planificar. Capacidad de integrar los conocimientos a la práctica.	Desarrollar sistemas de software integrando tecnologías para la solución de problemas, automatizando los procesos operativos, flujo de información y toma de decisiones en las organizaciones con un enfoque sistémico bajo estándares internacionales. Instrumentar proyectos tecnológicos bajo metodologías innovadoras, para el desarrollo de software y lograr la eficiencia de los procesos en las organizaciones, tomando en cuenta los requerimientos de los diversos sectores productivos. Coordinar equipos de trabajo inter y transdisciplinario, así como multiculturales para la gestión de proyectos computacionales supervisando las funciones y recursos de análisis funcional, orgánico y programación.



UNIDAD No. 1	Introducción a la Ingeniería de Software	Horas estimadas para cada unidad
CONTENIDOS		Evidencias de aprendizaje
Conceptuales	Aprendizajes esperados	
1.1. Conceptos Básicos 1.2. Evolución del Software 1.3. Características del Software 1.3.1.Elementos 1.3.2.Estructura 1.4. Importancia de la IS 1.5. Capas de la IS 1.6. Clasificación de los Software 1.7. Paradigmas de Ingeniería de Software 1.7.1.El enfoque estructurado 1.7.2.El enfoque orientado a objetos 1.7.3.Importancia de las herramientas CASE en la Ingeniería de software.	Identifica los paradigmas de ingeniería de software.	Problemas resueltos aplicando los paradigmas de ingeniería de software a situaciones reales.



UNIDAD No. 2	Modelos del Proceso de la Ingeniería de Software		Horas estimadas para cada unidad
CONTENIDOS			20
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
2.1. Un modelo General de proceso 2.1.1. Definición de actividad estructural 2.1.2. Identificación de un conjunto de tareas 2.1.3. Patrones del proceso 2.2. Modelos de proceso prescriptivo 2.2.1. Modelo en Cascada 2.2.2. Modelo del Proceso Incremental 2.2.3. Modelo de Proceso Evolutivo 2.2.4. Modelos Concurrentes 2.3. Modelos de proceso especializado 2.3.1. Desarrollo basado en componentes 2.3.2. Modelo de métodos formales 2.3.3. Desarrollo de software orientado a aspectos 2.4. El proceso unificado 2.4.1. Fases del proceso unificado	Aplica las etapas de un modelo de proceso de software.	Problema resuelto aplicando un modelo de desarrollo de software.	



UNIDAD No. 3	Análisis de los Requerimientos	Horas estimadas para cada unidad
		18
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
3.1. Ingeniería de requerimientos 3.2. Establecer las bases 3.2.1. Identificación de los participantes 3.2.2. Reconocer los múltiples puntos de vista 3.2.3. Trabajar hacia la colaboración 3.2.4. Hacer las primeras preguntas 3.3. Indagación de los requerimientos 3.3.1. Recopilación de los requerimientos en forma colaborativa 3.3.2. Despliegue de la función de calidad 3.3.3. Escenarios de uso 3.3.4. Indagación de los productos del trabajo 3.4. Requerimientos de las negociaciones 3.5. Validación de los requerimientos.	Aplica diferentes técnicas para la obtención de requerimientos de proyectos de software.	Reporte final de requerimientos para el diseño de un software.



UNIDAD No. 4	Diseño	Horas estimadas para cada unidad
		14
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
4.1. Diseño en el contexto de la ingeniería de software 4.2. Lineamientos y atributos de la calidad del software 4.3. Conceptos de diseño 4.3.1. Abstracción 4.3.2. Arquitectura 4.3.3. Patrones 4.3.4. División de problemas 4.3.5. Modularidad 4.3.6. Ocultamiento de la información 4.3.7. Independencia funcional 4.3.8. Refinamiento 4.3.9. Aspectos 4.3.10. Conceptos de diseño orientado a objetos 4.3.11. Clases de diseño	Aplica modelos, técnicas y herramientas para el diseño del software.	Diseño del software.

Contenidos procedimentales	Contenidos actitudinales
Identifica los paradigmas de ingeniería de software. Define las etapas de un modelo de proceso de software. Aplica diferentes técnicas para la obtención de requerimientos de proyectos de software. Aplica modelos, técnicas y herramientas para la etapa del diseño del software.	Responsabilidad en la entrega de sus trabajos. Honestidad en el manejo y presentación de la información. Diligente en el manejo de los datos. Disposición para trabajar en equipo. Rigor en el manejo de la información. Respeto a las propuestas de sus compañeros.



Metodología para la construcción del conocimiento

Actividades de aprendizaje con el docente	Actividades de aprendizaje autónomo
Exposición de temas.	Uso de herramientas diseño de software.
Exposición de ejercicios prácticos.	Investigación de casos.
Prácticas guiadas.	Solución de problemas.
Prácticas supervisadas.	

Evidencias de desempeño

Acreditación	Evaluación	Calificación
Cumplir con lo establecido en el Reglamento Escolar vigente. Entrega de evidencias de aprendizaje.	A final de cada unidad.	20% Problemas resueltos aplicando los paradigmas de ingeniería de software a situaciones reales. 20% Problema resuelto aplicando un modelo de desarrollo de software. 30% Reporte final de requerimientos para el diseño de un software. 30% Diseño del software.

FUENTES DE APOYO Y CONSULTA
BÁSICA

1. Pressman, S. (2010). Ingeniería del Software: un Enfoque Práctico. México: McGraw Hill.*
2. Cota, A. (2000). Ingeniería de Software: Soluciones Avanzadas. México: Oxford. *
3. Jacobson I., Booch G. & Rumbaugh J. (2006). El proceso unificado de desarrollo de software. Madrid: Addison Wesley
4. Kendall E. K., (2011). Análisis y Diseño de sistemas. México: Pearson Educación.*
5. Sommerville, I. (2005). Ingeniería del software, México: Prentice Hall.*

COMPLEMENTARIAS

1. Weitzenfeld A. (2005). Ingeniería de software orientada a objetos con UML, Java e Internet. México: Cengage Learning Editores.*
2. Scott, K. (2001). The Unified Process Explained. USA: Addison Wesley CE *

*La bibliografía con antigüedad mayor de cinco años contiene información relevante para el desarrollo de esta asignatura. Cabe destacar que son textos clásicos con ejemplos didácticos de fácil comprensión para el estudiante. Son difíciles de conseguir en el mercado, pero se encuentran en los catálogos de varias bibliotecas.

RESPONSABLE DEL DISEÑO

Elaborado por	Irlanda Yanet Ordoñez Sánchez, Fausto IV Flores Córdova.
Fecha de elaboración	20 de diciembre 2016.