



Nombre de la asignatura								Laboratorio de Diseño 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										
Asignaturas antecedentes	Asignaturas subsecuentes									Implícita
Ingeniería de Software	Ninguna									Conocimientos previos

PROPÓSITO DE LA ASIGNATURA										
Analizar y modelar proyectos de software aplicando el paradigma de ingeniería de software.										

COMPETENCIAS A DESARROLLAR										
Genéricas						Específicas				
Capacidad de aplicar los conocimientos en la práctica. Resolución de problemas. Trabajo en equipo.						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.				



UNIDAD No. 1	El modelo del proceso del software	Horas estimadas para cada unidad
		12
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
1.1. Conceptualización de tecnología orientada a objetos 1.2. Metodologías emergentes de desarrollo de software 1.3. Métodos de desarrollo de software orientado a objetos 1.4. El proceso de desarrollo unificado . RUP 1.5. El lenguaje de modelado unificado - UML	Aplica el modelo RUP en el desarrollo de software.	Solución de casos prácticos.

UNIDAD No. 2	Captura de requisitos	Horas estimadas para cada unidad
		16
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
2.1. Tipos de requisitos 2.2. Fuentes de datos para el análisis del sistema 2.3. Selección y diseño de instrumentos para la recopilación de información 2.4. Captura de requisitos candidatos 2.5. Selección de métodos de desarrollo 2.6. Modelo del negocio 2.7. Modelo del dominio 2.8. Validación de requerimientos 2.9. Definición de propuesta de solución	Identifica los requisitos para la construcción del software.	Reporte del modelo de negocio.



UNIDAD No. 3	Gestión de proyectos	Horas estimadas para cada unidad
		12
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
3.1. Actividades de gestión 3.2. Estudio de viabilidad del proyecto 3.3. Análisis Costo/beneficio 3.4. Gestión de riesgos 3.5. Planificación del proyecto 3.6. Calendarización del proyecto	Aplica los procedimientos para la gestión de proyectos.	Documento que describe el estudio de viabilidad del proyecto, reporte costo/beneficio, evaluación de riesgos y cronograma de actividades de un caso práctico.

UNIDAD No. 4	Modelado de Requisitos	Horas estimadas para cada unidad
		14
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
4.1. Modelos de casos de uso 4.1.1. Actores, Casos de uso, requerimientos funcionales y no funcionales 4.1.2. Prototipos para casos de uso 4.2. Documentación 4.2.1. Modelo de interfaces 4.2.2. Modelo del dominio del problema 4.2.3. Identificación de clases 4.2.4. Identificación de asociaciones 4.2.5. Identificación de atributos 4.3. Diccionario de clases 4.4. Identificación de módulos	Modela el dominio del problema identificando clases, asociaciones y atributos.	Modelo de casos de uso. Modelo de interfaz. Modelo del dominio del problema.



UNIDAD No. 5	Modelado de Análisis	Horas estimadas para cada unidad
		10
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
5.1. Modelos de análisis 5.2. Diagramas de secuencias 5.3. Diagramas de clases conceptuales 5.4. Realización de casos de uso 5.5. Presentación del proyecto final	Diseña el modelo de análisis.	Reporte del modelo de análisis.

Contenidos procedimentales	Contenidos actitudinales
Identifica los requerimientos. Analiza las necesidades. Analiza los riesgos. Plantea la solución. Elabora los diagramas correspondientes. Documenta el proyecto final.	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
Prácticas guiadas en el laboratorio de cómputo. Exposición de temas. Prácticas supervisadas.	Uso de herramientas CASE. Desarrollo de diagramas. Solución de problemas.



Evidencias de desempeño		
Acreditación	Evaluación	Calificación
Conforme a lo establecido en el Reglamento Escolar vigente. Documento del proyecto final.	Al término de cada unidad.	15% Solución de casos prácticos. 15% Reporte del modelo de negocio. 15% Documento que describe el estudio de viabilidad del proyecto, reporte costo/beneficio, evaluación de riesgos y cronograma de actividades de un caso práctico. 15% Modelo de casos de uso. 10% Modelo de interfaz. 10% Modelo del dominio del problema. 20% Reporte del modelo de análisis.

FUENTES DE APOYO Y CONSULTA	
BÁSICA	
1. Bruegge, B., & Dutoit, A. H. (2002). Ingeniería de software orientado a objetos. Prentice Hall.*	
2. Jacobson I., Booch G. & Rumbaugh J. (2006). El proceso unificado de desarrollo de software. Madrid: Addison Wesley*	
3. Laudon, K. C., & Laudon, J. P. (1996). Administración de los Sistemas de Información: Organización y tecnología. Prentice Hall Hispanoamericana.*	
4. Laudon, K. C., & Laudon, J. P. (2004). Sistemas de información gerencial: administración de la empresa digital. Pearson Educación.*	
5. Sommerville, I., & Galipienso, M. I. A. (2005). Ingeniería del software. Pearson Educación.*	
6. Rumbaugh, J., Booch, G., & Jacobson, I. (2000). El lenguaje unificado de modelado: manual de referencia*	
COMPLEMENTARIAS	
1. Lawrence, P. S. (2002). Ingeniería de Software, Teoría y Práctica. Editorial Prentice Hall.*	
2. Pressman, R. S. (1997). Ingeniería del Software: Un enfoque práctico. Mikel Angoar.*	
3. Whitten, J. L., Bentley, L. D., & Barlow, V. M. (2003). Análisis y diseño de sistemas de información. McGraw-Hill.*	

*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	Víctor Guillermo Castro, Elizabeth Torres Guillermo, Yeraldo de Jesús Delgado Montiel, Eloisa Mendoza Vázquez.
Fecha de elaboración	20 de diciembre 2016.