UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA Clave: 08MSU0017H Clave: 08USU4053W **FACULTAD DE INGENIERÍA** **PROGRAMA DEL CURSO:** **PROCESO Y CONSTRUCCIÓN DE SOFTWARE**	**DES:**	Ingeniería
	Programa(s) Educativo(s):	Ingeniería de Software
	Tipo de materia:	Obligatoria
	Clave de la materia:	IS301
	Semestre:	Tercero
	Área en plan de estudios:	Ciencias de la Computación e Informática
	Créditos	4
	Total de horas por semana:	4
	Teoría:	2
	Práctica	2
	Taller:	
	Laboratorio:	
	Prácticas complementarias:	
	Trabajo extra clase:	
	Total de horas semestre:	64
	Fecha de actualización:	Abril del 2008
	Materia requisito:	Ingeniería de Software y su computación II (cve. IS201)

Propósito del curso :
El alumno aprenderá a especificar, diseñar la descripción de un proceso de software desde una perspectiva particular comprendiendo las teorías y uso de herramientas y reglas para la formación de un software.

Al final del curso el estudiante:

- Enfoca a los procedimientos, técnicas y análisis para la creación de un software
- Explica las herramientas existentes para establecer las soluciones a las necesidades de implementar un software.
- Introduce conceptos en cuestión de calidad para las diferentes etapas o actividades del desarrollo de software.
- Define y diagnostica las fallas en un software a través de técnicas propuestas.

COMPETENCIAS (Tipo Y Nombre de la competencias que nutre la materia y a las que contribuye).	**DOMINIOS COGNITIVOS.** (Objetos de estudio, temas y subtemas)	**RESULTADOS DE APRENDIZAJE.** (Por objeto de estudio).
El curso promueve las siguientes competencias: **Competencias**	**UNIDAD I : TECNOLOGÍAS DE CONSTRUCCIÓN** 1.1. Diseño y uso de APIs	Define las técnicas que serán aplicadas a un software ejemplo,

Básicas: Solución de problemas Trabajo en equipo y liderazgo Comunicación **Competencias Profesionales**: Proyectos de Ingeniería Ingeniería de Proceso	1.2. Reuso de código y librerías 1.3. Aspectos de tiempo de ejecución de programación Orientado a Objetos 1.4. Parametrización y genéricos 1.5. Aseveraciones, Diseño por contrato y programación defensiva 1.6. Manejo de errores, excepciones y tolerancia a fallas 1.7. Técnicas de construcción basadas en estados y propulsadas por tablas 1.8. Configuración e internacionalización de tiempo de ejecución 1.9. Procesamiento de entrada basado en gramática (parsing) 1.10. Primitivas de concurrencia (semáforos, monitores, etc.) 1.11. Middleware (componentes y contenedores) 1.12. Métodos de construcción para software distribuido 1.13. Construcción de sistemas heterogéneos (hardware y software); codiseño software-hardware 1.14. Sintonización y análisis de desempeño	comprende la teoría descrita.
	UNIDAD II: HERRAMIENTAS DE CONSTRUCCIÓN 2.1. Ambientes de desarrollo 2.2. Constructores de GUI 2.3. Herramientas de prueba de unidad 2.4. Lenguajes orientados a aplicación (scripting, visual, específico de dominio, macros, markup etc)	Explica que es una estructura aplicada al desarrollo de un producto de software (un caso de uso como ejemplo) y los pasos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente

		para diferentes actividades que tienen lugar durante el proceso
	UNIDAD III: MÉTODOS FORMALES DE CONSTRUCCIÓN 3.1 Aplicación de máquinas abstractas (SDL, Paysley, etc.) 3.2 Aplicación de métodos y lenguajes de especificación (ASM, B, CSP, VDM, Z) 3.3 Generación automática de código a partir de especificación 3.4 Derivación de programas 3.5 Análisis de implementaciones candidatas 3.6 Mapeo de una especificación a diferentes implementaciones 3.7 Refinamiento	Demuestra las diferentes notaciones y técnicas para analizar sistemas, en el sentido de que están basados en alguna teoría matemática como, por ejemplo, la lógica para eliminar ambigüedades, y para depurar sistemas como ejemplos de uso.
	UNIDAD IV: CONCEPTOS DE PROCESOS 4.1 Temas y terminología 4.2 Infraestructura para el proceso de ingeniería de software (personal, herramientas, entrenamiento, etc) 4.3 Modelado y especificación de procesos de software 4.4 Medición y análisis de procesos de software 4.5 Mejoramiento del proceso de ingeniería de software (equipo, individual) 4.6 Análisis y control de calidad (prevención de defectos, revisión de procesos, métricas de calidad, análisis de causa raíz)	Identifica los modelos estructurados para las diferentes actividades como especificar, diseña e implementa la prueba de software (ejemplo de uso).
	UNIDAD V: IMPLEMENTACIÓN DE PROCESOS 5.1 Niveles de definición de procesos (organización, equipo de proyecto,	Desarrolla un modelo o esquema compartido entre el desarrollador y

	individual, etc) 5.2 Modelos de ciclo de vida (agile, heavyweight, waterfall, spiral, V-Model, etc) 5.3 Modelos y estándares del proceso de ciclo de vida (IEEE, ISO, etc.) 5.4 Proceso de software individual (modelo, definición, medición, análisis, mejora) 5.5 Proceso de equipo (modelo, definición, organización, medición, análisis, mejora) 5.6 Confección del proceso Requisitos del proceso de ciclo de vida del software (ISO/IEEE, Standard 12207)	cliente para tener una orientación de lo que se va a crear.

OBJETO DE ESTUDIO	**METODOLOGIA** (Estrategias, secuencias, recursos didácticos)	**EVIDENCIAS DE APRENDIZAJE.**
UNIDAD I : TECNOLOGÍAS DE CONSTRUCCIÓN UNIDAD II: HERRAMIENTAS DE CONSTRUCCIÓN UNIDAD III: MÉTODOS FORMALES DE CONSTRUCCIÓN UNIDAD IV: CONCEPTOS DE PROCESOS UNIDAD V: IMPLEMENTACIÓN DE PROCESOS	El profesor podrá utilizar las estrategias y secuencias que considere convenientes para que el estudiante logre el aprendizaje requerido. Los recursos didácticos podrán ser entre otros: Exposiciones, demostraciones, discusiones de grupo, preguntas y respuestas, revisiones de literatura, laboratorios, talleres, presentaciones por especialistas invitados de la industria.	**(**Durante el curso, al menos un programa debe ser realizado en equipo de al menos tres integrantes) Programas elaborados o modificados por el estudiante utilizando en forma separada o conjuntamente : Diseño y uso de APIs Reuso de código y librerías Aspectos de tiempo de ejecución de Programación Orientado a Objetos Parametrización y genéricos Aseveraciones, Diseño por contrato y programación

		defensiva Manejo de errores, excepciones y tolerancia a fallas Configuración e internacionalización de tiempo de ejecución Procesamiento de entrada basado en gramática (parsing) Primitivas de concurrencia (semáforos, monitores, etc.) Métodos de construcción para software distribuido Exámenes, presentaciones, reportes y/o foros donde el estudiante demuestre que distingue, contrasta o interpreta lo siguiente Técnicas de construcción basadas en estados y propulsadas por tablas Middleware (componentes y contenedores) Construcción de sistemas heterogéneos (hardware y software); codiseño software-hardware Exámenes,

		presentaciones, reportes y/o foros donde el alumno describa correctamente Sintonización y análisis de desempeño Un programa construido utilizando un ambiente de desarrollo Aplicación de máquinas abstractas (SDL, Paysley, etc.) Generación automática de código a partir de especificación Mapeo de una especificación a diferentes implementaciones Modelos de ciclo de vida (agile, heavyweight, waterfall, spiral, V-Model, etc) Modelos y estándares del proceso de ciclo de vida (IEEE, ISO, etc.) Proceso de software individual (modelo, definición, medición, análisis, mejora) Proceso de equipo (modelo, definición, organización, medición, análisis, mejora) Requisitos del proceso

		de ciclo de vida del software (ISO/IEEE, Standard 12207

FUENTES DE INFORMACIÓN (Bibliografía, Direcciones electrónicas)	**EVALUACIÓN DE LOS APRENDIZAJES** (Criterios e instrumentos)
1. McConnell Steve. (2004). *Code Complete: A Practical Handbook of Software Construction Microsoft Corporation*. (2ª Edition). USA. 2. Jacobson Ivar, Grady Booch, Rumbaugh James. (2000*). El Proceso Unificado de Desarrollo de Software*. (1ª Edición). ADDISON-WESLEY IBEROAMERICANA ESPAÑA, S.A. 3. Ezran Michel, Morisio Maurisio, Tully Colin. (2002). *Practical Software Reuse (Practitioner Series)*. (1a Edicion). Springer Verlag. 4. Mili Hafegh, Mili Ali, Yacoub Sherif, Addy Edward.(2001) *Reuse-Based Software Engineering: Techniques, Organizations, and Controls*. (1a Edicion). Wiley,USA. 5. Lutowski Rick. (2005). Software Require*ments: Encapsulation, Quality, and Reuse*. (1a Edicion). Auerbach, USA. 6. Steven Kelly y Juha-Pekka. (2008).*Domain-Specific Modeling: Enabling Full Code Generation*. (1a Edicion). Tolvanen. 7. Lanza Michele, Marinescu Radu.(2005) *Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems*. (1a Edicion). Springer Verlag, USA	**Criterios** El estudiante debe presentar satisfactoriamente al menos el 60 por ciento de las evidencias de aprendizaje solicitadas por el profesor en cada uno de los niveles de abstracción para considerar que obtuvo el nivel de competencia mínimo. **Instrumentos** Análisis de discusiones de grupo preguntas y respuestas análisis de reportes de revisión de literatura análisis de foros laboratorios talleres Exámenes Presentaciones

8. Neil G, Kluwer Jacobson (2004). *The In-System Configuration Handbook: A Designer's Guide to ISC* (1a Edicion). Academic Publishers USA.

9. Grune Dick, Ceriel J, H.Jacobs. (2008) *Parsing Techniques: A Practical Guide (Monographs in Computer Science).* (2ª Edicion). Springer,USA.

10. Burns Alan, Welling Andy, Barns John (1998). *Concurrency in Ada*. (1a Edicion). Cambridge University Press,Inglaterra.

11. Buschmann Frank, Henney Kevlin (2007). *Pattern-Oriented Software Architecture Volume 4: A Pattern Language for.(1a Edicion).* Distributed Computing, Wiley,USA.

12. Voelter Markus, Kircher Michael, Zdun Uwe. (2005). *Remoting Patterns: Foundations of Enterprise, Internet and Realtime Distributed Object Middleware (Wiley Software Patterns Series). (1a Edición).* Wiley,USA.

13. Cousin Doblado Javier José, Sanz Fernández Luis. (2000). *Medición Para La Gestion En La Ingeniería Software*. (1ª Edición). España.

14. J.Braude Eric. (2005). *Ingeniería de software : una perspectiva orientada a objetos* .(1ª Edición). España

15. Sommerville Ian, Galipienso Alfonso María Isabel, Martínez Botia. (2005). *Ingeniería del Software.* (7ª Edición). Pearson Educación Madrid.

16. Pressman Roger. (2010). *Software Engineering: A Practitioner's Approach*. (7ª Edicion). McGraw-Hill,USA.

17. McCourt Judith. (2005). *Understanding Interfaces: A Handbook of Human-Computer Dialogue*. (1a Edicion). Elsevier Science & Technology Books.

Cronograma Del Avance Programático

S e m a n a s

Objetos de estudio	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I. TECNOLOGÍAS DE CONSTRUCCIÓN																
II. HERRAMIENTAS DE CONSTRUCCIÓN																
III. MÉTODOS FORMALES DE CONSTRUCCIÓN																
IV. CONCEPTOS DE PROCESOS																
V. IMPLEMENTACIÓN DE PROCESOS																