

Universidad Nacional de San Agustín  
VICE RECTORADO ACADÉMICO  
SILABO

CODIGO DEL CURSO: CS390

**1 Datos Generales**

<b>FACULTAD :</b> Ingeniería de Producción y Servicios								
<b>DEPARTAMENTO :</b> Ingeniería de Sistemas e Informática				<b>ESCUELA :</b> Ciencia de la Computación				
<b>PROFESOR :</b>								
<b>TÍTULO :</b>								
<b>ASIGNATURA :</b> Ingeniería de Software II								
<b>PREREQUISITO:</b> CS290T		<b>CREDITOS:</b> 4			<b>Año:</b> 2010-1		<b>Total Horas:</b> 2 HT; 2 HP 2 HL	
					<b>Sem:</b> 6 <sup>to</sup> Semestre.			
<b>Horario</b>		Lun	Mar	Mie	Jue	Vie	Sáb	
<b>Total Semanal</b>								
<b>Aula</b>								

**2 Exposición de Motivos**

Los tópicos de este curso extienden las ideas del diseño y desarrollo de software desde la seguridad a la programación para abarcar los problemas encontrados en proyectos de gran escala. Es una completa de la Ingeniería de Software apreciada desde un punto de vista de Proyectos.

**2 Objetivo**

- Capacitar a los alumnos para formar parte y definir equipos de desarrollo de software que afronten problemas de envergadura real.
- Familiarizar a los alumnos con el proceso de administración de un proyecto de software de tal manera que sea capaz de crear, mejorar y utilizar herramientas y métricas que le permitan realizar la estimación y seguimiento de un proyecto de software.
- Crear , evaluar e implementar un plan de prueba para segmentos de código de tamaño medio , Distinguir entre diferentes tipos de pruebas , sentar las bases para crear, mejorar los procedimientos de prueba y las herramientas utilizadas con ese propósito.
- Seleccionar con justificación un apropiado conjunto de herramientas para soportar el desarrollo de un rango de productos de software.
- Crear , mejorar y utilizar los patrones existentes para el mantenimiento de software . Dar a conocer las características y patrones de diseño para la reutilización de software.
- Identificar y discutir diferentes sistemas especializados , crear , mejorar y utilizar los patrones especializados para el diseño , implementación , mantenimiento y prueba de sistemas especializados

**3 Contenido Temático 3 SE/Desarrollo de Sistemas Especializados.(12 horas)**

Objetivos Específicos
<ul style="list-style-type: none"> <li>▪ Identificar y discutir diferentes temas especializados.</li> <li>▪ Discutir el ciclo de vida y tópicos sobre el proceso de software en el ámbito de sistemas diseñados para un contexto especializado incluyendo sistemas que podrían tener que operar en un modo de operación graduado.</li> <li>▪ Seleccionar, con la justificación apropiada, métodos que darán como resultado el desarrollo eficiente y efectivo y el mantenimiento de temas de software especializado.</li> <li>▪ Dado un contexto específico y un conjunto de tópicos profesionales relacionados, discutir como, un ingeniero de software envuelto en el desarrollo de sistemas especializados, debe de responder a estos tópicos.</li> <li>▪ Sintetizar los temas técnicos centrales asociados con la implementación del crecimiento de sistemas especializados..</li> </ul>

**3 SE/Herramientas y Entornos de Software.(12 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Seleccionar con justificación un apropiado conjunto de herramientas para soportar el desarrollo de un rango de productos de software.</li> <li>▪ Analizar y evaluar un conjunto de herramientas en una área dada del desarrollo de software (ej: administración, modelamiento o pruebas).</li> <li>▪ Demostrar la capacidad para usar un rango de herramientas de software en soporte del desarrollo de un producto de software de tamaño medio.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Entornos de programación.</li> <li>▪ Análisis de requisitos y herramientas de modelamiento.</li> <li>▪ Herramientas de desarrollo de herramientas estáticas y dinámicas.</li> <li>▪ Herramientas de configuración.</li> <li>▪ Manejo de las herramientas de configuración.</li> <li>▪ Mecanismos de herramientas.</li> </ul> <p>[7], [2], [9], [10], [4], [3], [6]</p>

**3 SE/Validación y verificación de software.(12 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Distinguir entre validación de programas y verificación.</li> <li>▪ Describir el rol que las herramientas pueden jugar en la validación de software.</li> <li>▪ Distinguir entre los diferentes tipos y niveles de pruebas (unidad, integración, sistemas y aceptación) para productos de software de tamaño medio y el material relacionado.</li> <li>▪ Crear, evaluar e implementar un plan de prueba para segmentos de código de tamaño medio.</li> <li>▪ Encargarse, como parte de una actividad de equipo, de una inspección de un segmento de código de tamaño medio.</li> <li>▪ Discutir los temas concernientes a la prueba de software orientado a objetos..</li> </ul>	<ul style="list-style-type: none"> <li>▪ Distinción entre validación.</li> <li>▪ Abordajes estáticos.</li> <li>▪ Planeamiento de documentación para pruebas.</li> <li>▪ Diferentes tipos de pruebas: prueba de software humano-computadora, confiabilidad, seguridad con la especificación.</li> <li>▪ Fundamentos de pruebas: desde la creación del código y la generación de pruebas.</li> <li>▪ Técnicas de pruebas: pruebas de caja blanca y caja negra.</li> <li>▪ Semilla por defectos.</li> <li>▪ Unidad, integración, pruebas de sistemas de pruebas.</li> <li>▪ Prueba orientada a objetos de sistema.</li> <li>▪ Medidas de productividad de programa.</li> <li>▪ Verificación y validación: pruebas que no son computacionales, archivos de pruebas, de entrenamiento.</li> <li>▪ Defecto de historial, defecto de rastreo, para esas actividades.</li> <li>▪ Test de regresión.</li> <li>▪ Inspecciones, revisiones.</li> </ul> <p>[7], [2], [9], [10], [4], [11], [3], [6]</p>

**3 SE/Evolución del Software.(12 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"><li>▪ Identificar los temas principales asociados con la evolución del software y explicar su impacto sobre el ciclo de vida del software.</li><li>▪ Discutir los desafíos de mantener sistemas heredados y la necesidad de la ingeniería reversa.</li><li>▪ Delinear el proceso de pruebas de regresión y su rol en la administración del lanzamiento.</li><li>▪ Estimar el impacto de un cambio de requerimiento para un producto existente de tamaño medio.</li><li>▪ Desarrollar un plan para hacer reingeniería a un producto de tamaño medio como respuesta a un cambio de requerimientos.</li><li>▪ Discutir las ventajas y desventajas del reuso de software.</li><li>▪ Explotar las oportunidades para reusar software en un contexto dado.</li><li>▪ Identificar debilidades en un simple diseño dado y resaltar como las mismas pueden ser removidas a través de la reconstrucción (<i>refactoring</i>).</li></ul>	<ul style="list-style-type: none"><li>▪ Mantenimiento de software.</li><li>▪ Características del software mantenible.</li><li>▪ Reingeniería.</li><li>▪ Sistemas heredados.</li><li>▪ Reuso de software.</li></ul> <p>[7], [2], [9], [10], [4], [11], [8], [9], [5], [3], [6]</p>

**3 SE/Administración de Proyectos de Software.(12 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> <li>▪ Demostrar, involucrándose en un equipo de proyecto, los elementos centrales de la construcción y administración de un equipo.</li> <li>▪ Preparar un plan para un proyecto de software que incluye estimación de tamaño y esfuerzo, asignación de tiempos y tareas, asignación de recursos, control de configuración, administración de cambios, identificación y administración de los riesgos del proyecto.</li> <li>▪ Indicar un abordaje para tratar riesgos que ayudará a entregar el software a tiempo.</li> <li>▪ Comparar y contrastar los diferentes métodos y técnicas usados para asegurar la calidad de un producto de software.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Administración de recursos de equipos y tiempos y responsabilidades de software en la ejecución de un proyecto. Planes de ejecución.</li> <li>▪ Asignación de recursos en un proyecto.</li> <li>▪ Medición de riesgos y estimación de riesgos.</li> <li>▪ Análisis de riesgos de seguridad. a) Seguridad, sistemas de seguridad. b) El ciclo de vida.</li> <li>▪ Aseguramiento de la calidad de software. a) Métodos de aseguramiento de la calidad.</li> <li>▪ Administración de versiones y versiones de software. La versión 1.0.</li> <li>▪ Herramientas de administración de proyectos.</li> <li>▪ Modelos de administración de medidas de calidad.</li> </ul> <p>[7], [2], [9], [10], [3], [6]</p>

### 3 SE/Evaluación de riesgos.(6 horas)

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"><li>▪ Definir los conceptos de peligros y riesgos.</li><li>▪ Reconocer riesgos comunes de seguridad en al menos dos sistemas operativos.</li><li>▪ Describir las categorías de amenazas a sistemas de redes de computadores.</li><li>▪ Mostrar un abordaje sistemático para la tarea de identificar peligros y riesgos en una situación particular.</li><li>▪ Aplicar los principios básicos de manejo de riesgos en una variedad de escenarios incluyendo alguna situación relacionada con seguridad.</li></ul>	<ul style="list-style-type: none"><li>▪ Definición de términos: en seguridad, vulnerabilidad, amenazas, brechas de seguridad, peligros.</li><li>▪ El concepto de riesgo, identificación de peligros y riesgos.</li><li>▪ Análisis de riesgo incluyendo evaluación.</li><li>▪ Necesidad de un abordaje completo de sistema que incluya peligros asociados con herramientas.</li><li>▪ Riesgo y las tecnologías inmaduras.</li><li>▪ Análisis de costo beneficio.</li><li>▪ Principios del manejo de riesgos.</li></ul> <p>[7], [2], [9], [10], [4], [11], [8], [9], [5], [13], [6]</p>

### 4 Actividades

- Asignaciones
- Controles de Lectura
- Exposiciones

### 5 Recursos Materiales

- Apuntes del curso
- Libro(s) de la bibliografía

### 6 Metodología

- Clase Magistral.
- Taller didáctico.
- Social Constructivismo.
- Prácticas personales y en grupo.

### 7 Evaluación

La nota final ( $NF$ ) se obtiene de la siguiente manera:

**NE** Nota de Exámenes 60 %, esta nota se divide en

- Exámen Parcial 40 %
- Examen Final 60 %

**NT** Nota de Trabajos e Intervención en clase 40 %

$$NF = 0,6 * NE + 0,4 * NT$$

## Referencias

- [1] Vincenzo Ambriola. *Software Process Technology*. Springer, July 2001.
- [2] Bruce I. Blum. *Software Engineering: A Holistic View*. Oxford University Press US, 7th edition, May 1992.
- [3] R Conradi. *Software Process Technology*. Springer, March 2000.
- [4] Jessica Keyes. *Software Configuration Management*. CRC Press, February 2004.
- [5] Carlo Montangero. *Software Process Technology*. Springer, September 1996.
- [6] Flavio Oquendo. *Software Process Technology*. Springer, September 2003.
- [7] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 6th edition, March 2004.
- [8] John W. Priest and Jose M. Sanchez. *Product Development and Design for Manufacturing*. Marcel Dekker, January 2001.
- [9] Stephen R Schach. *Object-Oriented and Classical Software Engineering*. McGraw-Hill, January 2004.
- [10] Yingxu Wang and Graham King. *Software Engineering Processes: Principles and Applications*. CRC Press, April 2000.
- [11] Daniel R. Windle and L. Rene Abreo. *Software Requirements Using the Unified Process*. Prentice Hall, August 2002.

---

Docente del curso