

## PROGRAMA DE ASIGNATURA

<b>Nombre</b>	Ingeniería de Software II				
<b>Carrera</b>	Licenciatura en Ciencia de la Computación (4602)				
<b>Código</b>	22635				
<b>Créditos SCT-Chile</b>	9	<b>Trabajo Directo semanal</b>	6	<b>Trabajo Autónomo semanal:</b>	
<b>Nivel</b>	7				
<b>Requisitos</b>	Ingeniería de Software I (22630)				
<b>Categoría</b>	Obligatorio				
<b>Área de conocimiento según OCDE</b>	Ciencias Naturales				
<b>Descripción</b>	<p><b>Contribución al Perfil de Egreso</b></p> <p>Desempeños integrales:</p> <p><i>DI1. Elaborar modelos, diseños, e implementaciones para solucionar problemas o satisfacer necesidades relativas al procesamiento de datos, en instituciones u organismos de carácter público o privado, del ámbito de la producción o los servicios.</i></p> <p><i>DI3. Participar en la formulación y ejecución de proyectos de investigación, desarrollo e innovación, en su propia disciplina o en contextos multidisciplinares.</i></p> <p><i>DI4: Trabajar en equipo, de manera activa y contributiva, ejerciendo una comunicación efectiva, tanto en forma oral como escrita, en sintonía con las necesidades de aplicación y transferencia de los conocimientos disciplinares.</i></p> <p><i>DI5. Aprender de forma autónoma o guiada, para perfeccionarse tanto en ciencia de la computación como en el ejercicio profesional, en diferentes ámbitos de desarrollo.</i></p> <p>Conocimientos: Ingeniería del Software</p> <p>Habilidades y Destrezas Profesionales: pensamiento crítico, capacidad de resolución de problemas, capacidad para la organización y el procesamiento de información, capacidad para aplicar el pensamiento algorítmico, capacidad para realizar una representación eficiente y pertinente de información, sintetizar y elaborar modelos.</p> <p>Valores y actitudes: responsabilidad, ética en el uso y divulgación de la información, tolerancia y respeto interpersonal, trabajo en equipo, comunicación efectiva tanto oral, escrita como simbólica, uso eficiente de recursos informáticos.</p> <p><b>Resultado de aprendizaje general</b></p> <p>Desarrollar un proyecto de software en equipo aplicando técnicas de modelamiento de objetos bajo una arquitectura cliente-servidor distribuida.</p>				
	<b>Resultados de aprendizaje específicos</b>			<b>Unidades temáticas</b>	

	<ul style="list-style-type: none"> <li>Comprender el modelo n-capas para la construcción de software multiplataforma tanto a gran como en baja escala.</li> </ul>	I. Arquitecturas cliente/servidor en el desarrollo de software
	<ul style="list-style-type: none"> <li>Realizar el análisis y diseño de un sistema de software cliente/servidor aplicando técnicas de modelamiento de objetos a partir de requisitos.</li> </ul>	II. Análisis y Diseño del Software Cliente/Servidor
	<ul style="list-style-type: none"> <li>Justificar el resultado de la aplicación de un proceso de verificación y validación de software para detectar fallas y mejorar la calidad del software</li> </ul>	III. Verificación y Validación de Software
	<ul style="list-style-type: none"> <li>Desarrollar e implementar un sistema de software cliente/servidor de tres capas (presentación, negocio, datos) analizando los principales problemas en su mantención ante cambios en los requisitos o en la infraestructura tecnológica</li> </ul>	IV. Desarrollo y Mantención del Software
<p><b>Metodologías de enseñanza y de aprendizaje</b></p> <p>La asignatura de desarrollará en base a clases expositivas y participativas dialogadas, lectura de artículos y bibliografía relacionada con cada tema, guías de ejercicios para cada tema, realización de talleres grupales (trabajo y discusión), presentaciones orales de los estudiantes y debates sobre aspectos de actualidad relacionados con la asignatura.</p> <p>Las actividades prácticas (laboratorio) se realizarán utilizando el método ABP (Aprendizaje Basado en Proyectos) las cuales tienen por finalidad hacer que el alumno enfrente de manera adecuada un problema real, aplicando conocimientos adquiridos y desarrollando en equipo una solución de software de mediana escala, haciendo un uso intensivo de una base de datos relacional. El proyecto será analizado previamente por el profesor.</p> <p>Como apoyo al proceso de enseñanza/aprendizaje se dispone de un ambiente virtual en el LMS Moodle USACH (<a href="http://www.udesantiagoovirtual.cl/">http://www.udesantiagoovirtual.cl/</a>). En dicha plataforma el alumno encontrará material didáctico, guías de ejercicios, parte del material bibliográfico, como también los contenidos de las clases y de las actividades prácticas.</p>		
<p><b>Procedimientos de evaluación</b></p> <p>Las ponderaciones de cada actividad serán establecidas al comienzo del semestre por el profesor.</p> <p>Ponderación Teoría (50%): Pruebas escritas programadas (PEP), controles de lectura.</p> <p>Ponderación de trabajos prácticos (50%): proyecto de desarrollo grupal de software, trabajos prácticos relacionados a temáticas del curso.</p> <p>Las calificación de teoría como de trabajos prácticos se aprueban por separado.</p>		

**Bibliografía básica**

Pressman, Roger S. (2005). *Ingeniería del Software. Un Enfoque Práctico*. 6º edición. McGraw-Hill.

Sommerville Ian (2006). *Software Engineering*. 8th edition. Addison Wesley

Kendall, Kenneth E., Kendall, Julie E (2005). *Análisis y Diseño de Sistemas*, 6º Edición, Person Educación.

Aalto, J-M. & Jaaksi, A. (1999). *Tried & True Object Development: Practical Approaches with UML*. Cambridge University Press, Sigs Books,

Larman, Craig. (2005), *Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development*, 3º edition, Prentice Hall, Upper Saddle River.