

PRESIDENCIA ROQUE SÁENZ PEÑA, 22 de abril de 2013

RESOLUCIÓN N° 059/13 – C.D.C.B. y A.

VISTO:

El Expediente N° 01-2013-00871, iniciado por la Ing. Patricia Zachman, medio por el cual eleva el Programa del Curso Optativo “Técnicas Avanzadas De Diseño De Software” correspondiente a la Carrera Ingeniería en Sistemas de Información de la Universidad Nacional del Chaco Austral, para su aprobación; y

CONSIDERANDO:

Que el mencionado Programa se ajusta a los contenidos mínimos y carga horaria de la citada carrera;

Que se consideran adecuados los objetivos, métodos pedagógicos, métodos de evaluación, programa analítico y bibliografía que forman parte de la propuesta;

Que analizadas las actuaciones, el Consejo Departamental opina que lo solicitado se encuadra con lo establecido por el Reglamento Académico de Alumnos;
Lo aprobado en sesión de la fecha;

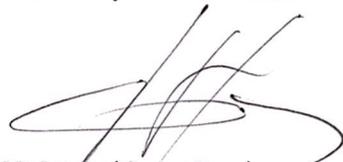
POR ELLO:

**EL CONSEJO DEPARTAMENTAL
DEL DEPARTAMENTO DE CIENCIAS BÁSICAS Y APLICADAS
DE LA UNIVERSIDAD NACIONAL DEL CHACO AUSTRAL
RESUELVE:**

ARTICULO 1º. Aprobar el Programa del Curso Optativo “**TÉCNICAS AVANZADAS DE DISEÑO DE SOFTWARE**” que corresponde a la carrera **Ingeniería en Sistemas de Información**, del Departamento de Ciencias Básicas y Aplicadas de la Universidad Nacional del Chaco Austral, y que como Anexo Único forma parte de la presente Resolución.

ARTÍCULO 2º. Regístrese, comuníquese a la **Ing. Patricia Zachman** y a las Áreas correspondientes. Cumplido, archívese.



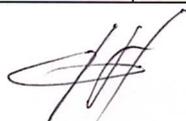

MG. ING. JOSÉ SERGIO FERNÁNDEZ
Director del Departamento
Ciencias Básicas y Aplicadas

 UNCAUS UNIVERSIDAD NACIONAL DEL CHACO AUSTRAL		CURSO OPTATIVO TÉCNICAS AVANZADAS DE DISEÑO DE SOFTWARE Resolución N° 059/13 – C.D.C.B.yA. ANEXO	
Departamento		Ciencias Básicas y Aplicadas	
Carga Horaria: 50 horas		Programa vigente desde: 2013	
Carrera		Año	Cuatrimestre
INGENIERÍA EN SISTEMAS DE INFORMACIÓN		Quinto	Primero
CORRELATIVA PRECEDENTE(*)		CORRELATIVA SUBSIGUIENTE(*)	
Asignaturas		Asignaturas	
Para cursar		Para rendir	
Regularizada	Aprobada	Aprobada	
-Ingeniería en Software	-Sistemas Operativos	-----	
OBJETIVOS:		<ul style="list-style-type: none"> • Brindar conocimientos acerca de las tendencias en modelados de software bajo el enfoque propuesto por el estándar del Object Management Group (OMG) denominado MDA (Model Driven Architecture). Dicho enfoque propone la generación de modelos de software independientes de las plataformas arquitectónicas donde se desplieguen. • Utilizar el lenguaje estándar UML para construir lenguajes específicos del dominio a través del uso de perfiles. • Conocer nuevo paradigma de programación. • Conocer las tendencias en reuso de diseño orientado a objetos basadas en sistemas de patrones arquitectónicos y patrones de diseño. • Desarrollar aplicaciones de software independientemente de las plataformas computacionales y lenguajes de programación. • Realizar herramientas automáticas que generen código fuente. 	
CONTENIDOS MÍNIMOS:		Arquitectura de Software a través del estudio y descripción de diferentes estilos arquitectónicos en términos abstractos. Reuso de diseño a través de arquitecturas genéricas. Técnicas de reuso de diseños genéricos orientados a objetos (frameworks). Diseño de Frameworks, Documentación e Instanciación de aplicaciones. Nuevas tendencias de reuso de diseño orientado a objetos basadas en sistemas de patrones arquitectónicos y patrones de diseño.	
MÉTODOS PEDAGÓGICOS:		La metodología de enseñanza de esta materia es teórico-práctica, el abordaje de los temas por parte del plantel docente invita a los alumnos a la realización continua de ejercicios prácticos. Los responsables de cátedra guiarán a los alumnos para la formación de equipos de trabajo, tratando de lograr homogeneidad de los integrantes respecto de carga de materias, disponibilidad de tiempo, conocimientos y experiencia estudiantil para disponer en el curso de equipos con similares condiciones. Cada alumno debe tener, en todas las clases, la carpeta de trabajos prácticos completa, para su trabajo y a disposición de los docentes.	
MÉTODOS DE EVALUACIÓN:		Trabajos Prácticos en Clase, en Laboratorio, en Tareas Extras. Participación del alumno en forma individual y/o grupal.	



///... RESOLUCIÓN N° 059/13 – C.D.C.B.yA.

<p>MÉTODOS DE EVALUACIÓN:</p>	<p>Informes sobre Análisis de Casos. El alumno debe concurrir al 100% de las clases en Laboratorios Prácticos y al 80 % de las Clases de Desarrollo Teórico de los Temas del Analítico. Aprobar los Laboratorios y Tareas Extras más los exámenes Parciales con un 60% como mínimo. Rendir Examen Final. Se tendrán en cuenta, además un conjunto de aspectos considerados como evaluación continua:</p> <ul style="list-style-type: none"> • Cumplimiento de fechas de presentación de los trabajos prácticos. • Calidad en el contenido y presentación de la carpeta de trabajos prácticos completa. • Asistencia. • Participación en clase, actitud y desempeño. • Claridad conceptual. • Creatividad e innovación. • Uso de herramientas. • Argumentación de ideas. • Búsqueda y consulta de material bibliográfico. • Retroalimentación de observaciones y errores. • Comportamiento en el equipo de trabajo.
<p>PROGRAMA ANALÍTICO:</p>	<p>UNIDAD 1: ARQUITECTURA Y DESARROLLO DE MODELOS.</p> <p>Estilos arquitectónicos: arquitecturas de flujo de datos, arquitecturas de componentes independientes, arquitectura de máquina virtual, arquitectura de repositorios, arquitectura en capas o niveles, arquitecturas MVC, otra. Principios de diseño - Arquitectura de Infraestructura. Core. Profiles: Alineación Arquitectónica entre UML y MOF. Arquitectura de Superestructura. Reuso de la Infraestructura. El Paquete kernel. Capas del Metamodelo.</p> <p>Desarrollo de software dirigido por modelos (MDD). La arquitectura MDA. Modelos y Transformaciones. Modelo independiente de la computación (CIM- Computation Independent Model). Modelo independiente de plataformas (PIM- Platform Independent Model). Modelo específico de la plataforma (PSM- Platform Specific Model). Modelo de implementación (ISM - Implementation Specific Model). Evolución de modelos y sus interrelaciones.</p> <p>UNIDAD 2: REUSABILIDAD DE SOFWTARE.</p> <p>Aspectos técnicos y no técnicos del reuso de software. Técnicas de reuso de diseños genéricos orientados a objetos (frameworks). Diseño de Frameworks. Documentación e Instanciación de aplicaciones. Nuevas tendencias de reuso de diseño orientado a objetos basadas en sistemas de patrones arquitectónicos y patrones de diseño.</p> <p>UNIDAD 3: ALMACENES DE DATOS, OLAP Y MINERÍA DE DATOS.</p> <p>Almacenes de Datos, OLAP y Minería de Datos: definición e interrelación. Arquitectura de un sistema de almacén de datos. Explotación de un almacén de datos: herramientas OLAP. Sistemas ROLAP y MOLAP. Carga y Mantenimiento de un Almacén de Datos. Diseño de un almacén de datos. Introducción a Minería de Datos (DM) Problemas tipo y aplicaciones. Relación de DM con otras</p>



///... RESOLUCIÓN N° 059/13 – C.D.C.B.yA.

<p>PROGRAMA ANALÍTICO:</p>	<p>disciplinas. Tipología de Técnicas de Minería de Datos. Sistemas Comerciales de Datos. Visualización. Técnicas de Minería de Datos. El Problema de la Extracción Automática de Conocimiento. Técnicas no supervisadas y descriptivas. Técnicas supervisadas y predictivas. Web Mining. Los Problemas de la Información No Estructurada. Extracción de Conocimiento a partir de Documentos HTML y texto. Extracción de Información semi-estructurada (XML).</p> <p>UNIDAD 4: ARQUITECTURAS EN ENTORNOS WEB Análítica web: concepto y orígenes, Análisis de logs. ¿Qué es un log?, Ventajas del análisis de logs. Aplicación de principios de usabilidad en el entorno web. Evaluación de la utilidad y otras herramientas tradicionales para recopilar información relativa a preferencias, hábitos y comportamientos de los usuarios en los sitios web. Análítica web en acción: herramientas. Análítica web y SEO (Search Engine Optimization)</p>
<p>BIBLIOGRAFÍA:</p>	<p>-Fontela C., (2008) Orientación a Objetos: Diseño y Programación. Editorial Nueva Liberia.</p> <p>-Fontela C., (2011) UML: Modelado De Software. Editorial Alfaomega Grupo Editor.</p> <p>-Fowler, M., (1997) Analysis Patterns, Reusable Object Models. Editorial Addison-Wesley Professional.</p> <p>-Gamma, E., (1995) Design patterns: Elements of reusable object-oriented software. Editorial Reading.</p> <p>-Hall M., (2000) Core Servlets and Java Server Pages. Editorial Prentice-Hall.</p> <p>-Warmer, J., (2003). Object Constraint Language, The: Getting Your Models Ready for MDA, Second Edition. Editorial Addison-Wesley.</p>

(*) Sujeto a cualquier modificación del Plan de Estudio




MG. ING. JOSÉ SERGIO FERNÁNDEZ
Director del Departamento
Ciencias Básicas y Aplicadas