

PRESIDENCIA ROQUE SÁENZ PEÑA, 20 de agosto de 2013

RESOLUCIÓN N° 098/13 – C.D.C.B. y A.

VISTO:

El Expediente N° 01-2013-01106, iniciado por el Ing. Javier AGUIRRE, medio por el cual eleva el Programa de la Asignatura: “Paradigmas de Programación” 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 de la Asignatura: “**PARADIGMAS DE PROGRAMACIÓN**” 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 al Ing. Javier AGUIRRE y a las Áreas correspondientes. Cumplido, archívese.




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



PARADIGMAS DE PROGRAMACIÓN

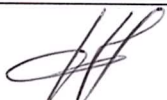
Resolución N° 098/13 – C.D.C.B.yA.

ANEXO

Departamento:		Ciencias Básicas y Aplicadas.	
Carga Horaria: 120 horas		Programa vigente desde: 2013	
Carrera		Año	Cuatrimestre
INGENIERÍA EN SISTEMAS DE INFORMACIÓN		Segundo	Segundo
CORRELATIVA PRECEDENTE (*)		CORRELATIVA SUBSIGUIENTE (*)	
Asignaturas		Asignaturas	
Para cursar		Para rendir	
Regularizada	Aprobada	Aprobada	
-Matemática Discreta. -Algoritmos y Estructuras de Datos.	-	-Matemática Discreta. -Algoritmos y Estructuras de Datos.	-Gestión de Datos. -Diseño de sistemas. -Sistemas Operativos.
DOCENTES:		Ing. AGUIRRE, Javier Raúl.	
OBJETIVOS:		<ul style="list-style-type: none"> Estudiar los aspectos fundamentales de los diferentes paradigmas de programación como proceso fundamental para la evaluación de los mismos. Desarrollar la capacidad práctica de aprendizaje de los elementos de cada paradigma, de su utilización e implementación a través de lenguajes específicos. Resolver problemas orientados a cada paradigma. Aplicar un paradigma con su respectivo lenguaje de programación en una problemática de interés para la Universidad. 	
CONTENIDOS MÍNIMOS:		Concepto de Paradigmas de Programación. Paradigmas fundamentales. Paradigma Funcional. Cálculo Lambda. Lenguajes de Programación Funcional. Paradigma Lógico. Lógica de Predicados de Primer Orden y Formas Restringidas. Regla Inferencia de Resolución. Lenguaje de Programación Lógica. Paradigma Orientado a Objetos. Conceptos Básicos. Clasificación, Clase y Objeto. Método y Mensaje. Clase Abstracta y Concreta. Herencia y Tipos de Herencia. Polimorfismo y Tipos de Polimorfismo en el Modelo de Objetos. Lenguajes de Programación Orientados a Objetos. Extensiones al Modelo Básico de Objeto en un Lenguaje Particular.	
MÉTODOS PEDAGÓGICOS:		La metodología comprende las siguientes actividades de: <ul style="list-style-type: none"> Exposición de clases teóricas buscando la interacción con los alumnos, mediante el dialogado de experiencias y conocimientos. Presentación de clases teóricas expositivas, vinculando con la realidad circundante en materia de problemas computacionales. Desarrollo de clases prácticas con resolución de ejercicios de programación que soporten los paradigmas desarrollados en clases teóricas. Atención de consultas sobre temas teóricos y prácticos. 	

///... RESOLUCIÓN N° 098/13 – C.D.C.B.yA.

MÉTODOS PEDAGÓGICOS	<ul style="list-style-type: none"> • Preparación y evaluación de Trabajos Prácticos • Valoración de los Procesos realizados. • Evaluación de los resultados obtenidos. • Realización de una aplicación de interés institucional, que se desarrollará fuera del horario de clase, despertando el interés, la creatividad y la profundización del lenguaje aplicado, mediante tutorías y clases de consulta.
MÉTODOS DE EVALUACIÓN:	<p>a) Forma de evaluación y controles.</p> <ul style="list-style-type: none"> • Revisión y Aprobación de los TP. • Participación individual efectiva en trabajo en equipo. <p>b) Instancias de aprobación</p> <p>b.1) Para Promocionar: Asistencia 80 % Aprobación de los Trabajos Prácticos, en tiempo y forma. Aprobación de los dos Parciales, con promedio de calificación mayor o igual a 8, y el Trabajo de Desarrollo, con calificación mayor o igual a 8.</p> <p>b.2) Para Regularizar: Asistencia 70 % Aprobación de los Trabajos Prácticos, en tiempo y forma. Aprobación de los dos Parciales, con promedio de calificación mayor o igual a 6, y el Trabajo de Desarrollo, con calificación mayor o igual a 6.</p> <p>La modalidad de los parciales será escrito teórico – práctico.</p> <p>Aprobación del Final Teórico, en caso que corresponda, el mismo tendrá carácter oral en la teoría – defensa del trabajo y escrito-en máquina para cuestiones prácticas.</p>
PROGRAMA ANALÍTICO:	<p>UNIDAD 1:INTRODUCCIÓN Qué es un Paradigma. Qué es un Paradigma de Programación. Criterios para el estudio y análisis de lenguajes. Importancia de estudiar conceptos comparados en lenguajes de programación para la formación de criterios.Características de lenguajes de programación representativos de cada programa. Historia de los lenguajes de Programación. Datos de mercado.</p> <p>UNIDAD 2: PARADIGMA DE PROGRAMACIÓN IMPERATIVA. Peso histórico y cuantitativo de los lenguajes imperativos. Orígenes y evolución. Puntos relevantes y aspectos críticos. Elementos de la programación Imperativa. Abstracción de la Función y Procedimiento.</p> <p>UNIDAD 3:CONCEPTOS COMPARADOS EN LENGUAJES DE PROGRAMACIÓN Valores y Tipos. Tipos Simples y Compuestos. Sistemas de Tipos. Equivalencia estructural. Verificación de tipos de la forma estática y dinámica. Principio de Completitud de Tipo. Lenguajes con sistemas fuertemente tipados. Tipos de monomórficos. Inferencia de tipos monomórficos. Sobrecarga. Polimorfismo. Interferencia de tipos polimórficos. Tipos genéricos. "BINDING": enlace o resolución de vínculos entre entidades y atributos. Enlace y ámbito. Lenguajes con estructura de bloques. Alcance y visibilidad. Declaraciones. Alcance de las declaraciones. Ligamiento estático y dinámico. Mecanismos de Abstracción. Mecanismos de abstracción en datos y en control. Tipos (abstractos) de datos. Parámetros. Mecanismos de copia y Mecanismos por definición. Orden de evaluación de parámetros. Evaluación impaciente y en orden Normal. Funciones estrictas y no estrictas. Evaluación perezosa (Lazy). Variables y Actualizaciones. Variables compuestas. Tiempo de Vida de una variable.</p>



///... RESOLUCIÓN N° 098/13 – C.D.C.B.yA.

<p>PROGRAMA ANALÍTICO:</p>	<p>Variables globales y locales. Variables de pila y variables persistentes. Conclusiones. Balance de características y propiedades de lenguajes de programación. Aportes oportunos de cada paradigma. Mención de otros paradigmas. Nuevos enfoques: fronteras entre paradigmas.</p> <p>UNIDAD 4: PARADIGMA DE PROGRAMACIÓN ORIENTADA A OBJETOS. Impacto en el enfoque Orientado a Objetos. Conceptos básicos de la Programación en Grande. Encapsulamiento. Ocultamiento. Modularidad. Jerarquía. Polimorfismo. Objetivos de la Orientación a Objetos. Ventajas. Programación estructurada vs. POO. Definiciones de términos. Objetos. Clases. Ocultación de información y visibilidad. Instanciación. Notación de objetos. Mensajes. Herencia simple y múltiple. Redefinición. Sobrecarga. Clases abstractas. Variables de instancia de Clases. Métodos de instancia y de Clases. Funciones virtuales. Polimorfismo. Ligadura dinámica. C++, Java.</p> <p>UNIDAD 5: PARADIGMA DE PROGRAMACIÓN DE LÓGICA. Programación Declarativa. Principios del Paradigma lógico. Concepción de problemas. Características particulares. Representación formal. Espacio de Búsqueda. Tipos de soluciones. Búsqueda. Búsqueda en grados y árboles. Algoritmo de Control. Dirección de la búsqueda. Encadenamiento hacia adelante. Encadenamiento hacia atrás. Métodos de búsqueda. Primero en profundidad. Primero en amplitud. Ventajas y desventajas. Backtraking cronológico. Lógica Proposicional. Lenguajes de representación. Sintaxis, Semántica. Formulas Bien Formadas. Interpretación Lógica. Consecuencia Lógica. Métodos de prueba. Inferencia Lógica. Cláusulas. Litera Proposicional. Cláusula Proposicional. Cláusula Proposicional de Horn. Regla de Resolución proposicional. Refutación y Deducción. Mundo Cerrado. Árbol de resolución. Árbol SLD. Negación en Programación Lógica. Lógica de Predicados. Lógica de Predicados de Primer Orden. Sintaxis y Semántica. Cláusulas de Skolem. Cláusulas de Horn. Sustitución y Unificación. UMG. Principio de Resolución de Robinson. Regla de Resolución de Predicados. Notación Prolog. Lenguaje de Programación de Prolog. Hechos. Consultas. Variables. Variables anónimas. Conjunciones y Disyunciones. Aritmética. Reglas. Backtraking y Recursión. El predicado Cut y Fail. Listas. Representación. Unificación. Ejemplos.</p> <p>UNIDAD 6: PARADIGMA DE PROGRAMACIÓN FUNCIONAL. Antecedentes Históricos. Importancia de los lenguajes funcionales. Lenguajes funcionales puros e híbridos. El cálculo lambda. Fundamentos. Expresiones lambda. Sintaxis. Convenciones sintácticas. Variables: ámbito, libre y ligadas. Lambda Reducciones. λ-reducción o λ-conversión. λ-reducción. Ordenes de Reducción propiedad de Confluencia y Terminación. Estructura de los Lenguajes Funcionales. Definición y llamada funciones. Forma normal. Modos de evaluación. Normal y lazy vs. Eager. Polimorfismo. Transparencia Referencial. Recursividad. Lenguaje de programación Haskell. Valores y Tipos. Funciones. Funciones de orden superior. Estructuras de Datos infinitas. Expresiones Case y Patternmatching.</p>
<p>BIBLIOGRAFÍA:</p>	<p>Bibliografía Obligatoria:</p> <ul style="list-style-type: none"> • Watt David, Programming Languages Concepts and Paradigms, Hoare Series Editor, Prentice Hall, 1990. • HuthMichael and Mark Ryan, Logic in Computer Science. Cambridge University Press, 2004. • Vidart Jorge y TasistroAlvaro, Programación Lógica y Funcional, 2005 • Cow Brad y J. Novobilski Andrew, Programación Orientada a Objetos. Un enfoque evolutivo. Addison-Wesley/Diaz de Santos, 1993. • Schneider Gregory R. Andrews and Fred B., Concepts and Notation for

///... RESOLUCIÓN N° 098/13 – C.D.C.B.yA.

BIBLIOGRAFÍA:

Concurrent Programming, Marzo 1983.

- Práctico – Apuntes de Cátedra. Compiladores e Intérpretes. Programming

Languages:

<http://burks.bton.ac.uk/burks/software/langs/index.htm>

Programación Funcional Hugs Online: <http://cvs.haskell.org/Hugs/>

Bibliografía Complementaria:

- Carlo Ghezzi & Mehdi Jazayeri, Programming Languages Concepts, John Wiley & Sons, Inc, 1987.
- Burns Alan & Wellings Andy, Sistemas de Tiempo real y lenguajes de Programación, Tercera Edición, Addison Wesley. 2003.
- Cardelli Luca & Wegner Peter, On understanding Types, Data Abstractions and Polymorphism. ACM Computing Surveys, Vol. 17, N4, Dec. 1985.
- Horowitz Ellis, Fundamentals of Programming Languages. Computer Science Press Inc. Springer-Verlag, 1983.
- Alfonseca Manuel, Programación Orientada a Objetos. Teorías y Técnicas OPP para desarrollo de software, Ediciones Anaya. 1992.

(*) Sujeto a cualquier modificación del Plan de Estudio




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