

Universidad Nacional de San Agustín
VICE RECTORADO ACADÉMICO
SILABO

CODIGO DEL CURSO: CS343

1 Datos Generales

FACULTAD : Ingeniería de Producción y Servicios						
DEPARTAMENTO : Ingeniería de Sistemas e Informática				ESCUELA : Ciencia de la Computación		
PROFESOR :						
TÍTULO :						
ASIGNATURA : Lenguajes de Programación						
PREREQUISITO: CS211T,CS210T		CREDITOS: 4		Año: 2010-1		Total Horas: 2 HT;
				Sem: 7 ^{mo} Semestre.		2 HT 2 HP 2 HL
Horario		Lun	Mar	Mie	Jue	Vie
Total Semanal						Sáb
Aula						

2 Exposición de Motivos

A pesar de que los algoritmos han sido diseñados y escritos por lo menos desde el tiempo de Euclides los últimos cincuenta años (desde el desarrollo de la computadora digital) los métodos de expresar el objeto de un extenso estudio. En la actualidad existen distintos paradigmas de programación, ciertos de los cuales están en uso activo, muchos más en existencia y aún más por ser diseñados. El propósito de esta asignatura es una introducción a los principios del estudio de la programación, y brindar los fundamentos básicos para que el estudiante pueda brindar un estudio exhaustivo de los principios del diseño de los lenguajes de programación es que el estudiante se convierta en un mejor programador. Adicionalmente este curso es útil si se necesita saber acerca de que lenguaje de programación usar para un proyecto, o si alguna vez necesita diseñar su propio lenguaje de programación.

2 Objetivo

- Capacitar a los estudiantes para entender los lenguajes de programación desde diferentes tipos de vista, según el modelo subyacente, los componentes fundamentales presentes en todo lenguaje de programación y como objetos de programación formales dotados de una estructura y un significado según diversos enfoques.

Objetivos Específicos

- Reconocer el desarrollo de los lenguajes de programación
- Identificar los paradigmas de programación existentes
- Discutir entre los paradigmas y establecer sus ventajas
- Establecer otros criterios de clasificación de los lenguajes de programación
- Diferenciar entre lenguajes estructurales, basados en léxico hasta el semántico
- Identificar el concepto de programación orientada a objetos
- Diferenciar entre lenguajes de programación y la interpretación de programas
- Reconocer como funciona un programa a nivel de código

3 Contenido Temático 3 El desarrollo histórico de los lenguajes de programación (4 horas)**3 Lenguajes Imperativos (4 horas)**

Objetivos Específicos	Contenidos
<ul style="list-style-type: none"> ▪ Identificar los principios de la programación imperativa ▪ Determinar las bases del imperativismo:secuencialidad, selección y repetición ▪ Aprender cómo los lenguajes imperativos manejan datos y asigna valores ▪ Aprender el concepto de ortogonalidad en un lenguaje de programación ▪ Diseña e implementa un programa en un lenguaje de programación imperativo 	<ul style="list-style-type: none"> ▪ Introducción ▪ Manejo de datos y tipos ▪ Asignaciones y Expresiones ▪ Flujos de control ▪ Componentes de un programa imperativo ▪ Ejemplos de programas imperativos <p>[1], [3], [4]</p>

3 Orientación a Objetos (8 horas)

Objetivos Específicos	Contenidos
<ul style="list-style-type: none">▪ Identificar los principios básicos en los cuales se basa la programación orientada a objetos▪ Analiza como pasar del dominio de un problema a un modelado orientado a objetos▪ Aprende como representar a nivel de lenguaje y a nivel de abstracción un caso problema▪ Aprende la sintaxis de un lenguaje de programación orientado a objetos puro▪ Implementa un programa en lenguaje de programación orientado a objetos▪ Analiza los distintos tipos de herencia que presentan los lenguajes de programación orientados a este paradigma y examina sus ventajas y desventajas	<ul style="list-style-type: none">▪ Introducción a los principios de la programación orientada a objetos▪ Conceptos básicos: Clases, Herencia y Polimorfismo▪ Binding Dinámico▪ Semántica referencial▪ Ejemplos de programas orientados a objetos <p>[1], [3], [4]</p>

3 Lenguajes Funcionales (4 horas)

Objetivos Específicos	Contenidos
<ul style="list-style-type: none">▪ Reconocer los principios fundamentales del paradigma funcional▪ Comparar las ventajas de la orientación funcional sobre otros esquemas▪ Analiza el concepto de funciones y lo aplica en la solución de problemas▪ Establece la facilidad del uso de lenguajes funcionales para casos en estructuras de datos y recursividad▪ Investiga sobre el cálculo lambda▪ Diseña e implementa programas en algún tipo de lenguaje funcional	<ul style="list-style-type: none">▪ Introducción a los lenguajes funcionales▪ Definición de función▪ Listas▪ Tipos y Polimorfismo▪ Funciones de orden superior▪ Lazy Evaluation▪ Ejemplos de programas funcionales <p>[1], [3], [4]</p>

3 Lenguajes Lógicos (4 horas)

Objetivos Específicos	Contenidos	Horas
<ul style="list-style-type: none">▪ Comprender el modo de operación de los lenguajes lógicos orientados al logro de metas▪ Analizar el encadenamiento hacia adelante o hacia atrás▪ Aprender un lenguaje orientado al paradigma lógico▪ Diseñar e implementar programas en lenguajes de programación orientados a objetos	<ul style="list-style-type: none">▪ Principios▪ Cláusulas de Horn▪ Variables Lógicas▪ Relaciones y Estructuras de Datos▪ Control del orden de búsqueda▪ Ejemplos de programas basados en el paradigma lógico <p>[1], [3], [4]</p>	

3 Otros Paradigmas (2 horas)

Objetivos Específicos	Contenidos	Horas
<ul style="list-style-type: none">▪ Identifica otros paradigmas presentes en nuestro medio▪ Analiza si es que los paradigmas estudiados son novedosos o solamente una consecuencia de los principales paradigmas analizados▪ Critica la máquina de Von Neumann en base a los conocimientos de su arquitectura▪ Diseña e implementa programas en un lenguaje de programación basado en los paradigmas estudiados	<ul style="list-style-type: none">▪ Programación Paralela▪ Programación Distribuida▪ Crítica a la máquina de Von Neumann <p>[1], [2], [3], [4]</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] Henry E. Bal and Dick Grune. *Programming Language Essentials*. Addison-Wesley, 1994.
- [2] Conrad Mueller. *Addressing: The root of all programming evils*. Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004.
- [3] Robert W. Sebesta. *Concepts of Programming Languages*. Addison-Wesley, 2005.
- [4] Ravi Sethi. *Programming Languages: Concepts and Constructs, Second Edition*. Addison Wesley Publishing Company, 1996.

Docente del curso