

	UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. <i>Creada por Ley N°:1.009/96 del 03/12/96</i> Facultad de Ingeniería	
PROGRAMA DE ESTUDIOS		

Materia:	Diseño de Compiladores	Semestre:	Décimo	
Ciclo:	Ingeniería Informática			
Código:	120			
Horas Semanales:	Teóricas:			4
	Prácticas:			
	Laboratorio:			2
Horas Semestrales:	Teóricas:			68
	Prácticas:			
	Laboratorio:	34		
Pre-Requisitos:	Algoritmos y Estructuras de Datos II, Microprocesadores y Lenguaje de Ensamblador, Paradigmas de la Programación			

I- OBJETIVOS GENERALES

Los objetivos de esta materia son desarrollar en el alumno las capacidades de:

Conocer, comprender y manejar conceptos y técnicas vinculados con la construcción de compiladores.

II - OBJETIVOS ESPECIFICOS



Desarrollar en el alumno capacidades como:

1. Comprender el funcionamiento interno de los lenguajes y compiladores, sus componentes y las herramientas existentes para el desarrollo de los mismos.
2. Comprender los conceptos básicos sobre Autómatas finitos aplicados a los compiladores.
3. Diseñar Gramáticas de lenguajes
4. Implementar un compilador dado una Gramática de un lenguaje
5. Manejar conceptos de optimización de código en los compiladores
6. Resolver problemas de programación aplicando los conceptos dados.

Algunas capacidades adicionales:

1. Implementación de soluciones usando librerías como flex o cup (o similares, según el lenguaje usado)
2. En esta materia se suele dar en java, sin embargo queda a criterio de la Dirección de Carrera cuál lenguaje es mejor utilizar en coordinación con las otras materias.

Aprobado por _____	Actualización No.: _____	_____ Sello y Firma	Página 1 de 4
Fecha: _____	Resolución No.: _____		
	Fecha: _____		

	UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. <i>Creada por Ley N°:1.009/96 del 03/12/96</i> Facultad de Ingeniería	
	PROGRAMA DE ESTUDIOS	

III. CONTENIDOS PROGRAMÁTICOS

Unidad I

Introducción

1. Conceptos
2. Componentes de un Compilador.
3. Programas relacionados con el compilador
4. Procesos de traducción
5. Arranque automático y portabilidad

Unidad II

Análisis léxico

1. Analizador léxico (Scanner)
2. Expresiones regulares
3. Autómatas finitos No determinísticos
4. Autómatas finitos Determinísticos
5. Herramientas para análisis léxico. Lex, Flex y derivados

Unidad III

Analizador sintáctico descendente

1. Analizador Sintáctico (Parser)-
2. Gramáticas formales (Gramáticas libres de contexto). Backus Naur Form
3. Parsing. Top down y Bottom up parsing
4. Ejercicios de gramáticas de lenguajes, Ambigüedades
5. Abstract Syntax Trees (Parsing trees)
6. Herramientas para análisis sintáctico. Yacc y derivados

Unidad IV

Análisis semántico

1. Análisis Semántico - definición
2. Scope-checking y Tabla de símbolos.
3. Type-checking

Aprobado por _____ Fecha: _____	Actualización No.: _____ Resolución No.: _____ Fecha: _____	_____ Sello y Firma	Página 2 de 4
--	---	------------------------	---------------

	UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. <i>Creada por Ley N°:1.009/96 del 03/12/96</i> Facultad de Ingeniería	
	PROGRAMA DE ESTUDIOS	

Unidad V

Generación y representación

1. Generación de Representación Intermedia
2. Runtimeenvironments
3. Three Access code
4. Generación de TAC
5. Optimización de RI
6. Optimizaciones Locales
7. Optimizaciones Globales

Unidad VI

Generación de código

1. Generación de código
2. Asignación de registros
3. Recolección de basura
4. Optimización

IV. METODOLOGIA

En las clases teóricas se explican los conceptos del tema dado presentando ejemplos de problemas a resolver, enmarcando los lineamientos para una posible solución y permitiendo la participación de los alumnos para contribuir al razonamiento de esta solución.

En las clases de Laboratorio se realizan experiencias demostrativas de conceptos teóricos, y trabajos de programación aplicando los conceptos teóricos (en un lenguaje como Java).

También se realizan varios proyectos de programación aplicando los conceptos teóricos y las prácticas del laboratorio dados en el transcurso de la materia.

V- CRITERIOS DE EVALUACION

Conforme al Reglamento Académico y Reglamento de Cátedra vigentes.

Aprobado por _____ Fecha: _____	Actualización No.: _____ Resolución No.: _____ Fecha: _____	_____ Sello y Firma	Página 3 de 4
--	---	------------------------	---------------

	UNIVERSIDAD NACIONAL DE ITAPUA – U.N.I. <i>Creada por Ley N°:1.009/96 del 03/12/96</i> Facultad de Ingeniería	
	PROGRAMA DE ESTUDIOS	

VI. BIBLIOGRAFÍA

Appel, A. W. (2002). Modern Compiler Implementation in Java. Cambridge University Press.

Louden, K. C. (2004). Construcción de compiladores: principios y práctica. Thomson.

Sethi, R., &Ullman, J. D. (1998). Compiladores: principios, técnicas y herramientas. Pearson Educación.

Contenidos según guía de la ACM 2013:

ProgrammingLanguages (PL), Software Engineering (SE), Algorithms and Complexity (AL)

AL Basic AutomataComputability and Complexity

AL Advanced Automata Computability and Complexity

PL Program Representation

PL Language Translation and Execution

PL Syntax Analysis

PL Compiler Semantic Analysis

PL Code Generation

PL Runtime Systems

PL Static Analysis

SE Software Design

Aprobado por _____ Fecha: _____	Actualización No.: _____ Resolución No.: _____ Fecha: _____	_____ Sello y Firma	Página 4 de 4
--	---	------------------------	---------------