



A. IDENTIFICACIÓN

CARRERA:	INGENIERÍA INFORMÁTICA
ASIGNATURA:	DISEÑO DE COMPILADORES
SIGLA:	INF 3631
DURACIÓN:	Un semestre académico (20 semanas)
HORAS SEMANALES:	Teóricas: 4, Prácticas: 1, Laboratorio: 1, TOTAL: 6
PLAN DE ESTUDIOS:	2011

B. CONTRIBUCIÓN AL PERFIL

Objetivos:

Al finalizar la asignatura, el estudiante debe ser capaz de emplear un método para el desarrollo y construcción de un compilador, que comprenda un analizador léxico, un analizador sintáctico, analizador semántico, generador de código intermedio, optimizador de código, generador de código objeto, manejador de errores y tabla de símbolos. Así mismo debe ser capaz de aplicar técnicas utilizadas en el desarrollo de compiladores a otro tipo de problemas similares.

Unidades de competencia:

- Habilidad para aplicar conocimientos matemáticos científicos y de ingeniería.
- Habilidad para identificar y solucionar problemas de ingeniería.
- Habilidad para usar técnicas y destrezas y herramientas necesarias para la práctica de la ingeniería.
- Desarrollar aplicaciones empleando lenguajes de programación.
- Habilidad para proponer soluciones algorítmicas.

C. CONTENIDO PROGRAMÁTICO

Contenido mínimo:

Introducción a los Compiladores.- Lenguajes Formales y Gramáticas de los Lenguajes.- Análisis Léxico.- Análisis Sintáctico, Análisis Semántico.- Generación de Código, Optimización de Código

Contenido analítico:

Tema 1: Introducción a los Compiladores.

- 1.1 Compiladores.
- 1.2 Programas relacionados con los compiladores.
- 1.3 Las Fases de un Compilador.
- 1.4 Herramientas para la Construcción de Compiladores.

Tema 2: Lenguajes formales y gramática de los lenguajes.



- 2.1 Conceptos de lenguajes.
- 2.2 Nociones de gramática.
- 2.3 Clasificación de Chomsky.
- 2.4 Gramática de contexto libre.
- 2.5 Árboles de análisis gramatical.
- 2.6 Árboles sintácticos abstractos.
- 2.7 Ambigüedad.
- 2.8 Notaciones BNF y EBNF.
- 2.9 Propiedades de las gramáticas de contexto libre.

Tema 3: Análisis léxico.

- 3.1 El proceso del análisis léxico.
- 3.2 Expresiones regulares.
- 3.3 Autómatas finitos.
- 3.4 De las expresiones regulares a los autómatas finitos deterministas.
- 3.5 Implementación de un Analizador Léxico.

Tema 4: Análisis Sintáctico.

- 4.1 El proceso del Análisis Sintáctico.
- 4.2 Análisis Sintáctico Descendente mediante método descendente recursivo.
- 4.3 Análisis Sintáctico LL (1).
- 4.4 Conjuntos Primero y siguiente.
- 4.5 Recuperación de errores en analizadores sintácticos descendentes.
- 4.6 Generadores de analizadores sintácticos.
- 4.7 Ejemplo de un analizador sintáctico.

Tema 5: Análisis Semántico.

- 5.1 Atributos y gramáticas con atributos.
- 5.2 Algoritmos para cálculos de atributos.
- 5.3 La tabla de símbolos.
- 5.4 Tipos de datos y verificación de Tipos.
- 5.5 Equivalencia de Expresiones de Tipos.
- 5.6 Ejemplo de un analizador semántico.

Tema 6: Generación de Código.

- 6.1 Código Intermedio.
- 6.2 Técnicas básicas de generación de código.
- 6.3 Generación de código de referencias de estructuras de datos.
- 6.4 Generación de código de sentencias de control y expresiones lógicas.
- 6.5 Generación de código de llamadas de procedimientos y funciones.

Tema 7: Optimizador de Código.

- 7.1 Principales fuentes para la optimación de código.
- 7.2 Tipos de Optimización.



7.3 Técnicas de Optimización de código.

D. REFERENCIAS BIBLIOGRÁFICAS

- [1] Louden K. C. (2004). **Construcción de Compiladores. Principios y práctica.** Thomson,
- [2] Aho A. V., Lam M. S., SEIT V., Ullman J.D. (2008). **Compiladores, principios, técnicas y herramientas.** Pearson. Addison-Wesley.
- [3] Sanchí Llorca F.J., Galán Pascual C. (1988). **Compiladores: Teoría y Construcción.** Paraninfo.