

# FACULTAD DE INGENIERÍA

## ÁREA DE COMPUTACIÓN E INFORMÁTICA



**Nombre de la materia :** COMPILADORES E  
INTÉRPRETES A

**Clave de la materia:**

**Clave Facultad:** 2406

**Clave U.A.S.L.P.:**

**Clave CACEI:** CI

**Nivel del Plan de Estudios:** IC: 7, II: 7 **No. de créditos:**  
10

**Horas/Clase/Semana:** 5 **Horas totales/Semestre:** 80

**Horas/Práctica (y/o Laboratorio):** 0

**Prácticas complementarias:**

**Trabajo extra-clase Horas/Semana:** 5

**Carrera/Tipo de materia:** Común del Área Obligatoria

**No. de créditos aprobados:**

**Fecha última de Revisión Curricular:** 12/ 03/ 2010

**Materia y clave de la materia requisito:** GRAFOS (2133)

### PROPÓSITO DEL CURSO

Dado que las computadoras sólo pueden interpretar lenguaje de máquina (Binario), no hay lenguaje de programación de alto nivel que no contenga un traductor, sea compilador o intérprete. Actualmente no es común que se cree o se le dé mantenimiento a alguno. Sin embargo, esto no indica que no haya desarrollos con base

en metodologías de traductores enfocadas a la resolución de problemas particulares de Software. La mayoría de implementaciones se hacen con lenguajes específicos (menos extensos) y se utilizan en la generación de herramientas para soporte de software de sistemas.

### OBJETIVO DEL CURSO

El alumno comprenderá los conceptos y comparará diferentes técnicas utilizadas en el diseño de compiladores e intérpretes de Software de Sistemas de uso general (lenguajes de programación). Elaborará en cada tema el programa de Computadora que realice la

función de cada una de las partes de un compilador: Analizador léxico, Analizador sintáctico. Así mismo la integración de dichas partes las cuales conjuntan el traductor.

### CONTENIDO TEMÁTICO

#### I.- Introducción.

Tiempo Estimado: 4 horas

##### Objetivo:

El alumno analizará y conceptualizará los sistemas de traducción, sus partes, la interacción entre ellos y sus aplicaciones.

##### 1.1 Traductores

- Compilador
- Intérpretes

##### 1.2 Códigos de máquina.

- Puro
- Aumentado
- Virtual

##### 1.3 Formato de código de máquina.

- Lenguaje Ensamblador
- Código binario
- Código de memoria con imagen

##### 1.4 Partes esenciales de un compilador(definición).

- Analizador léxico
- Analizador sintáctico
- Analizador semántico

#### II.- Autómatas.

Tiempo estimado: 2 horas

##### Objetivo:

El alumno comprenderá la importancia de la teoría de las gramáticas formales en el desarrollo de los lenguajes de programación.

##### 2.1 AFN

##### 2.2 AFD

##### 2.3 Pushdown

#### III.- Gramáticas formales

Tiempo estimado: 13 horas

##### Objetivo:

El alumno comprenderá e implementará las diferentes metodologías del análisis léxico.

##### 3.1 Clasificación de gramáticas

##### 3.2 Representación de gramáticas y lenguajes especiales

##### 3.3 Parser

- Parsing Top-Down
- Parsing Bottom-Up

- 3.4 Ambigüedad.
- 3.5 Factorización
- 3.6 Recursión
- 3.7 Operaciones aplicadas a lenguajes.
- 3.8 Expresiones regulares
- 3.9 Conjuntos no regulares.

IV.- Analizador léxico.

Tiempo estimado: 14 horas

Objetivo:

El alumno comprenderá e implementará las diferentes metodologías del análisis léxico.

- 4.1 Transformación de expresiones regulares a un AFN
- 4.2 Conversión de AFN a AFD por construcción de subconjuntos
- 4.3 Minimización de estados.
- 4.4 Construcción de un AFD a partir de una ER

V.- Analizador Sintáctico.

Tiempo estimado: 34 horas

Objetivo:

El alumno analizará, conceptualizará, e implementará alguna técnica del analizador sintáctico.

- 5.1 Analizador sintáctico predictivo
- 5.2 Analizador sintáctico por precedencia de operadores
- 5.3 Analizadores sintácticos LR
- 5.4 SLR
- 5.5 LR canónico
- 5.6 LALR

VI.- Herramientas LEX y YACC

Tiempo estimado: 13 horas

Objetivo:

Conocer y realizar implementaciones en base a las herramientas LEX y YACC para generar traductores.

- 6.1 Introducción.

**METODOLOGÍA**

Exposición de los temas, trabajos complementarios, análisis y discusión sobre la implementación de métodos, toda implementación se llevará a cabo utilizando un lenguaje de programación visual especificado por el

profesor. El alumno también conceptualizará e implementará de un traductor en forma individual, para lo cual tendrá que hacer el ligado entre las herramientas LEX y YACC y el lenguaje visual .

**EVALUACIÓN**

De acuerdo con el reglamento de exámenes, se realizarán cinco exámenes parciales en las fechas establecidas por el calendario de actividades de la Facultad. La calificación

final es el promedio de los cinco exámenes parciales. Para poder presentar exámenes es requisito realizar los programas considerados en cada parte del contenido.

**BIBLIOGRAFÍA**

Bibliografía Básica

Compiladores, Principios, Técnicas y Herramientas  
Aho, Sethi, Ulman  
Ed. Addison-Wesley Iberoamericana, 1990

Bibliografía Complementaria

The theory and practice of compiler writing  
Tremblay, Sorenson  
Ed. Mc Graw-Hill International Edition, 1985

Introducción a la teoría de Autómatas, lenguajes y computación  
John E. Hopcroft  
Jeffrey D. Ullman  
CECSA 1993

Fundamentos de Compiladores  
Karen A. Lemone  
CECSA, 1996

Teoría de autómatas y lenguajes formales  
Dean Kelley  
Prentice-Hall, 1995

Crafting a compiler with C  
Fischer, Charles N./ Le Blanc, Richard J, Jr.  
The Benjamin/Cummings Publishing Company, Inc.  
1991

Apuntes de la materia de Compiladores e Intérpretes  
Ing. Miguel Ángel Botello Aragón  
Facultad de Ingeniería, 2004

Lex & Yacc : Manual de Referencia.  
Ing. Miguel Angel Botello Aragón  
Facultad de Ingeniería, 2000