



**COMPILADORES**

7º

09

Asignatura

Clave

Semestre

Créditos

Ingeniería Eléctrica

Ingeniería en Computación

Ingeniería en Computación

División

Departamento

Carrera en que se imparte

**Asignatura:**

Obligatoria

Optativa

**Horas:**

Teóricas

Prácticas

**Total (horas):**

Semana

16 Semanas

**Modalidad:** Curso.

**Asignatura obligatoria antecedente:** Lenguajes Formales y Automatas.

**Asignatura obligatoria consecuyente:** Ninguna.

**Objetivo(s) del curso:**

El alumno aplicará técnicas y herramientas de desarrollo de compiladores para elaborar software de base (intérpretes, compiladores y otros traductores) obteniendo un software que optimice tanto memoria como desempeño. Además el alumno podrá discernir entre los diferentes traductores existentes para elaborar software eficiente y adecuado al tipo de problema a resolver.

**Temario**

NÚM.	NOMBRE	HORAS
1.	Elementos para el estudio de los compiladores	8.0
2.	Análisis léxico	8.0
3.	Análisis sintáctico	6.0
4.	Análisis sintáctico descendente	8.0
5.	Análisis sintáctico ascendente	10.0
6.	Traducción dirigida por sintaxis	8.0
7.	Organización de memoria en tiempo de corrida	6.0
8.	Generación de código intermedio y análisis semántico	6.0
9.	Optimización de código	6.0
10.	Generación de código	6.0
		72.0
	Prácticas de laboratorio	0.0
	Total	72.0

**COMPILADORES**

(2 / 5)



**1 Elementos para el estudio de los compiladores**

**Objetivo:** El alumno describirá las etapas en el proceso de compilación, sin importar el tipo de traductor de que se trate.

**Contenido:**

- 1.1 Clasificación de los traductores
- 1.2 Estructura de un compilador
- 1.3 Entornos de ejecución
- 1.4 Consideraciones léxicas
- 1.5 Criterios generales de sintaxis
- 1.6 Problemas de traducción
- 1.7 Especificación de tipos y estructuras de datos

**2 Análisis léxico**

**Objetivo:** El alumno construirá un analizador léxico a partir de la definición de clases de componentes léxicos.

**Contenido:**

- 2.1 Funciones de un Analizador Léxico
- 2.2 Identificación de Clases léxicas
- 2.3 Estructura de las Tablas de símbolos
- 2.4 Manejo de errores léxicos
- 2.5 Programación de un Analizador Léxico (scanner)
- 2.6 Generación automática de Analizadores Léxicos

**3 Análisis sintáctico**

**Objetivo:** El alumno explicará a detalle la etapa del análisis sintáctico en el proceso de compilación así como las gramáticas idóneas para definición de la estructura de los lenguajes de programación.

**Contenido:**

- 3.1 Gramáticas idóneas para análisis sintáctico.
- 3.2 Representación de sintaxis con gramáticas BNF.
- 3.3 Clasificación de los analizadores sintácticos

**4 Análisis sintáctico descendente**

**Objetivo:** El alumno construirá un analizador sintáctico descendente a partir de una gramática adecuada para este tipo de análisis sintáctico.

**Contenido:**

- 4.1 Gramáticas LL.
- 4.2 Construcción de la Tabla de Parser para un análisis descendente
- 4.3 Manejo de errores sintácticos



#### 4.4 Construcción de un analizador sintáctico descendente recursivo

### 5 Análisis sintáctico ascendente

**Objetivo:** El alumno construirá un analizador sintáctico ascendente a partir de una gramática adecuada para este tipo de análisis sintáctico.

**Contenido:**

- 5.1 Gramáticas LR
- 5.2 Analizador SLR(1)
- 5.3 Analizador LR(1)
- 5.4 Analizador LALR(1)
- 5.5 Detección y recuperación de errores.
- 5.6 Generadores de analizadores de sintaxis LALR(1). YACC

### 6 Traducción dirigida por la sintaxis

**Objetivo:** El alumno modificará gramáticas y realizará la traducción dirigida por la sintaxis en analizadores descendentes y ascendentes.

**Contenido:**

- 6.1 Gramáticas de traducción
- 6.2 Manejo de la tabla de símbolos
- 6.3 Traducción dirigida por sintaxis en analizadores descendentes
- 6.4 Traducción dirigida por sintaxis en analizadores ascendentes

### 7 Organización de memoria en tiempo de corrida

**Objetivo:** El alumno analizará las estructuras para manejar la memoria en el momento de ejecución del programa.

**Contenido:**

- 7.1 Gramáticas de traducción
- 7.2 Diferentes tipos de organizaciones
- 7.3 Organización de pila o 'stack'
- 7.4 Organización de 'heap'
- 7.5 Paso de parámetros

### 8 Generación de código intermedio y análisis semántico

**Objetivo:** El alumno describirá los diferentes tipos de código intermedio y el análisis semántico

**Contenido:**

- 8.1 Atributos y gramáticas con atributos
- 8.2 Algoritmos de manejo de atributos
- 8.3 Lenguajes intermedios
- 8.4 Revisión de tipos y declaraciones
- 8.5 Generación de código intermedio de diferentes sentencias



### 9 Optimización de código

**Objetivo:** El alumno utilizará las diferentes técnicas para optimizar código

**Contenido:**

- 9.1 Principales fuentes para la optimización
- 9.2 Optimización de bloques básicos
- 9.3 Grafos de flujo

### 10 Generación de código

**Objetivo:** El alumno utilizará las técnicas para generar y optimizar código objeto

**Contenido:**

- 10.1 La máquina objeto
- 10.2 Técnicas básicas de generación de código

#### Bibliografía básica:

#### Temas para los que se recomienda:

AHO, A. V., SETHI, Ravi, ULLMAN, J.D. <i>Compiladores. Principios, técnicas y herramientas.</i> México Addison-Wesley Iberoamericana, 2000	<b>Todos</b>
LOUDEN, Kenneth C. <i>Compiler Construction. Principles and Practice</i> U.S.A. Thompson Learning, 1997	<b>Todos</b>
PITTMAN, Thomas, PETERS, James <i>The Art of Compiler Design: Theory and Practice</i> Englewood cliffs New Jersey USA Prentice Hall, 1992	<b>Todos</b>
TREMBLAY, Jean-Paul, SORENSON, Paul G. <i>The Theory and practice of compiler writing</i> U.S.A. Mc. Graw-Hill, 1985	<b>Todos</b>

**COMPILADORES**

(5/5)

**Bibliografía complementaria:**

BENNETT, J. P.  
*Introduction to Compiling Techniques. A first Course using  
 Ansi C, LEX and YACC*  
 U.S.A.  
 Mc. Graw-Hill. Book Company Europe, 1996

**Todos**

KAPLAN, Randy M.  
*Constructing Language Processor for Little Languages*  
 Portland  
 Wiley, 1994

**Todos**

LEVINE, Jhon R, MASON, Tony, BROWN, Doug  
*Lex and Yacc*  
 2a. Edition  
 U.S.A.  
 O'Reilly, 1992

**2 y 5**

MAK, Ronald  
*Writing Compilers and Interprets*  
 2a. Edición  
 U.S.A.  
 Willey, 1996

**Todos**

PRATT, T. W, ZELKOWITZ, M. V.  
*Lenguajes de Programación. Diseño e Implementación.*  
 México  
 Prentice Hall, 1998

**1**

SCOTT, Michael L.  
*Programming Language Pragmatics*  
 U.S.A.  
 Morgan Kaufmann, 2000

**Todos****Sugerencias didácticas:**

Exposición oral

Exposición audiovisual

Ejercicios dentro de clase

Ejercicios fuera del aula

Seminarios

Lecturas obligatorias

Trabajos de investigación

Prácticas de taller o laboratorio

Prácticas de campo

Otras

**Forma de evaluar:**

Exámenes parciales

Exámenes finales

Trabajos y tareas fuera del aula

Participación en clase

Asistencias a prácticas

Otras

**Perfil profesiográfico de quienes pueden impartir la asignatura**

Egresado de la carrera de Ingeniero en Computación o afín; recomendable con grado de Maestro o Doctor.  
 Conocimientos y experiencia en el diseño y construcción de lenguajes de programación.