



Universidad Autónoma "Gabriel René Moreno"
**FACULTAD DE INGENIERIA EN CIENCIAS
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



PROGRAMA ANALÍTICO DE ASIGNATURA

1. IDENTIFICACION DE LA MATERIA

NOMBRE DE LA ASIGNATURA:	COMPILADORES
PRE-REQUISITOS	: INF319 Y INF310
SIGLA Y CODIGO	: INF-329
NIVEL	: Sexto Semestre
HORAS	: 6 (4 HT, 2 HP)
CREDITOS	: 5
REVISADO EN	:

2. JUSTIFICACION

Aunque es probable que pocos especialistas de la información construyan o llegue a dar mantenimiento o un compilador para un lenguaje de programación importante, el educando puede aplicar con provecho las ideas y técnicas que se estudia en esta asignatura al diseño del software de propósito general. Por ejemplo, las técnicas de concordancia de cadena (strings) para construir analizadores léxicos también se usa en editores de texto, es sistemas de consultas de la información, y en programas de reconocimiento de patrones.

La teórica de los lenguajes formales, es considerada un prerrequisito indispensable para el desarrollo de un compilador, pues con ellas se pueden definir y probar matemáticamente las construcciones que guiaran al compilador.

3. OBJETIVOS DE LA ASIGNATURA

3.1. OBJETIVO GENERAL.

Al finalizar la asignatura, el estudiante será capaz de:

Comprender el proceso de compilación, desde el principio hasta el final, utilizando los instrumentos matemáticos provisto por la teoría de los "Lenguajes Formales".



Universidad Autónoma "Gabriel René Moreno"
**FACULTAD DE INGENIERIA EN CIENCIAS
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



3.2. OBJETIVOS ESPECIFICOS

- Relacionar los lenguajes regulares con los Autómatas Finitos, reduciendo de allí la implementación de un analizador léxico, la primera fase de un compilador.
- Establecer el nexo matemático-algorítmico que exista entre una gramática independiente del contexto o BNF con la especificación del analizador sintáctico de un compilador.
- Especificar implementar el análisis semántico de un compilador a través de un grafo dirigido a cíclico o GDA.
- Desarrollar la implementación de un generador de código intermedio, usando esquema de traducción a código de tres direcciones.
- Utilizando traducciones ensamblador por cada instrucción intermedia.

4. CONTENIDO MINIMO

Conceptos introductorios: Definición y Fases de Compilador.
Análisis Léxico.
Análisis Sintáctico
Código Intermedio: Código de tres direcciones.
Generación de Código Objeto –final–.

5. UNIDADES DEL PROGRAMA ANALITICO

UNIDAD I. LAS FASES DE UN COMPILADOR.

Tiempo: 3 horas

Objetivo

Conocer la arquitectura de un compilador entendiendo la relación e interacción entre sus módulos.

Contenido

- 1.1. Diferencias entre un compilador y un intérprete.
- 1.2. Las etapas de análisis y síntesis.
- 1.3. Explicación de las fases de cada etapa.
 - 1.3.1. El analizador léxico (Lexer, Scanner).
 - 1.3.2. El analizador sintáctico (Parser).
 - 1.3.3. El analizador semántico.
 - 1.3.4. El generador de código intermedio.
 - 1.3.5. El optimador de código.



Universidad Autónoma "Gabriel René Moreno"
**FACULTAD DE INGENIERIA EN CIENCIAS
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



- 1.3.6. El generador de código absoluto.
- 1.3.7. La tabla de símbolos.
- 1.3.8. Módulo de errores.

UNIDAD II. ANÁLISIS LÉXICO.

Tiempo: 24 horas

Objetivo

Aplicar el conocimiento de las expresiones regulares, en la construcción de un Analizador Léxico, la primera fase de un compilador.

Contenido

- 2.1 Función del analizador léxico.
- 2.2 Especificación de los componentes léxicos (Token).
 - 2.2.1 Uso de expresiones regulares en la especificación de los Tokens.
 - 2.2.2 Diagramas regulares.
 - 2.2.3 Los atributos de los Tokens.
- 2.3 Diagrama de transiciones (dt): una extensión de un AFD.
- 2.4. Acciones semánticas en un dt.
- 2.5. Implantación de un analizador léxico.

UNIDAD III. ANALISIS SINTACTICO Y SEMANTICO.

Tiempo: 24 horas

Objetivo

Implementar un Parser Predictivo acompañado de su Analizador Semántico.

Contenido

- 3.1 La función del analizador sintáctico o Parser.
- 3.2 Gramáticas independientes del contexto o BNF.
- 3.3 Arboles de análisis sintácticos.
- 3.4 Gramáticas LL (1).
- 3.5 Implantación de un traductor predictivo.
- 3.6 Gramáticas LR.
- 3.7 El analizador semántico como corrutina del Parser.
 - 3.7.1 Conversiones de tipo.
 - 3.7.2 Evaluación de tipos en tiempos de ejecución.



Universidad Autónoma "Gabriel René Moreno"
**FACULTAD DE INGENIERIA EN CIENCIAS
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



3.7.3 Uso de grafos dirigidos acíclicos en la comprobación de tipos.

UNIDAD IV. GENERACION DE CODIGO INTERMEDIO

Tiempo: 21 horas

Objetivo

Desarrollar la implementación de un generador de código intermedio, usando esquemas de traducción a código de tres direcciones.

Contenido

- 4.1 Conceptualización.
- 4.2 El código de tres direcciones.
- 4.3. Representación usando tercetos.
- 4.4 Representación usando cuádruplas.
- 4.3 Esquemas de traducción del programa fuente al lenguaje intermedio.
 - 4.3.1 Expresiones Aritméticas.
 - 4.3.2 Expresiones booleanas.
 - 4.3.3 La construcción `if..then..else.`
 - 4.3.4 La construcción `while.`
 - 4.3.4 La construcción `switch o case.`

UNIDAD V. GENERACION DE CODIGO OBJETO FINAL

Tiempo: 21 horas

Objetivo

Generar código fuente assembler, a partir de la representación intermedia.

Contenido

- 5.1 Conceptualización.
- 5.2 Formato de una archivo .ASM
- 5.3. Asignación de la memoria para el tiempo de ejecución.
- 5.4 Conversión de las Cuádruplas o Tercetos a sentencias ASM
 - 5.4.1 Asignaciones
 - 5.5.2 Factores booleanos.
 - 5.5.3 Rutinas de Biblioteca
- 5.5 Uso del enlazador (Linker).



Universidad Autónoma "Gabriel René Moreno"
**FACULTAD DE INGENIERIA EN CIENCIAS
 DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



6. METODOLOGIA

COMPONENTE	MÉTODO
Temas teóricos conceptuales	Clases magistrales y multimedia.
Temas Prácticos	Ejemplificación en computadora
Proyectos	Consulta bibliográfica, internet y presentaciones.

7. CRONOGRAMA

SEMANA	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ACTIVIDADES																		
Unidad I																		
Unidad II																		
Proyecto #1																		
Primer Parcial																		
Unidad III																		
Unidad IV																		
Unidad V																		
Proyecto #2																		
Segundo Parcial																		
Examen Final																		

8. SISTEMA DE EVALUACION

La nota final de 100 puntos, se obtiene de la siguiente manera:

2 exámenes parciales	60%
Proyectos	15%
Examen Final	25%



Universidad Autónoma "Gabriel René Moreno"
**FACULTAD DE INGENIERIA EN CIENCIAS
DE LA COMPUTACIÓN Y TELECOMUNICACIONES**



9. BIBLIOGRAFIA

- Aho, R. y Ullman, J. *Compiladores: Principios, Técnicas y Herramientas*, Addison-Wesley, 1999.
- Aho, R. y Ullman, J. *The Theory of Parsing, Translation and Compiling. Vols 1 y 2*. PrenticeHall, 1972.
- Appel, A. y Ginsburg, M. *Modern Compiler Implementation in C*. Cambridge University Press, 1998.
- Muchnick, S. *Advanced Compile Design and Implementation*. Morgan Kaufmann Publishers, 2007.
- Pittman, T. y Peters, J. *Art of Compiler Design. The Theory and Practice*. Prentice Hall, 2007.
- Teufel, B.; Schmidt, S. y Teufel, T. *Compiladores: Conceptos fundamentales*. Addison-Wesley. 2005.