

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS  
LICENCIATURA EN SISTEMAS COMPUTACIONALES**

**Área de formación:** Disciplinaria  
**Unidad académica:** Compiladores  
**Ubicación:** Quinto semestre  
**Clave:** 1990  
**Horas semana-mes:** 5  
**Horas teoría:** 3  
**Horas práctica:** 2  
**Unidades CONAIC:** 58.67  
**Prerrequisitos:** Ninguno  
**Horas de infraestructura:** 2  
**Créditos:** 8

### **PRESENTACIÓN**

El estudio de los compiladores e intérpretes es una disciplina de suma importancia en la formación profesional del estudiante dado a que la interacción entre el ser humano y la computadora es más fácil debido a los avances en las siguientes ramas del software: lenguajes de programación, sistemas operativos e interfaces.

El conocimiento de la teoría y las técnicas que se utilizan en la construcción de compiladores le permitirá al estudiante aplicar los conocimientos en lenguajes de programación, arquitectura de computadoras, teoría matemática de la computación, algoritmos e ingeniería de software. Así también conocer las diferentes herramientas automáticas que permiten la construcción de compiladores e intérpretes.

### **OBJETIVO GENERAL**

Al terminar el curso el estudiante aplicará los principios, técnicas, estructuras y herramientas para la construcción de compiladores.

### **UNIDAD I.- INTRODUCCIÓN A LA TEORIA DE COMPILADORES**

**TIEMPO APROXIMADO:** 5 Horas

**OBJETIVO DE LA UNIDAD:** Conocer los fundamentos y herramientas para el diseño de un compilador.

#### **CONTENIDO**

- 1.1 Conceptos generales
- 1.2 Etapas de un compilador
- 1.3 Compiladores e Intérpretes
- 1.4 Herramientas automáticas para generar compiladores

### **UNIDAD II.- ANÁLISIS LÉXICO**

**TIEMPO APROXIMADO:** 15 Horas

**OBJETIVO DE LA UNIDAD:** Construir un analizador léxico

### **CONTENIDO**

- 2.1 Definición de análisis léxico
- 2.2 Expresiones regulares
- 2.3 Autómatas.
  - 2.3.1 Autómatas no determinísticos
  - 2.3.2 Autómatas determinísticos
- 2.4 Matrices de transición
- 2.5 Tabla de símbolos
- 2.6 Programación de un analizador léxico básico
- 2.7 Diferentes herramientas automáticas para generar analizadores léxicos

### **UNIDAD III.- ANÁLISIS SINTÁCTICO**

**TIEMPO APROXIMADO:** 20 Horas

**OBJETIVO DE LA UNIDAD:** Aplicar las técnicas de desarrollo y construcción de analizadores sintácticos.

### **CONTENIDO**

- 3.1 Diagramas de sintaxis
- 3.2 Gramáticas
  - 3.2.1 Métodos descendentes
    - 3.2.1.1 Retroceso
    - 3.2.1.2 Predictivos
      - 3.2.1.2.1 Recursivo
      - 3.2.1.2.2 No recursivo
  - 3.2.2 Métodos ascendentes
    - 3.2.2.1 Desplazamiento/reducción
- 3.3 Algoritmos para el análisis sintáctico
- 3.4 Analizadores LL
- 3.5 Analizadores LR y LALR
- 3.6 Manejo de errores
- 3.7 Programación de un analizador sintáctico
- 3.8 Diferentes herramientas automáticas para generar analizadores sintácticos

### **UNIDAD IV.- ANÁLISIS SEMÁNTICO**

**TIEMPO APROXIMADO:** 20 Horas

**OBJETIVO DE LA UNIDAD.** Aplicar las técnicas de notaciones para asociar reglas semánticas en la construcción de gramáticas.

### **CONTENIDO**

- 4.1 Traducción dirigida por sintaxis
- 4.2 Gramáticas de atributo
- 4.3 Desarrollo de intérpretes
- 4.4 Analizadores LL y LR con símbolos de acción
- 4.5 Generación de código intermedio
- 4.6 Manejo de la tabla de símbolos
- 4.7 Programación de un analizador semántico

## UNIDAD V.- GENERACIÓN DE CÓDIGO

**TIEMPO APROXIMADO:** 20 Horas

**OBJETIVO DE LA UNIDAD.** Aplicar las técnicas y herramientas automáticas para la generación de código.

### CONTENIDO

- 5.1 Aspectos del diseño de un generador de código
- 5.2 La máquina objeto
- 5.3 Generación de directivas
- 5.4 Generación de código de los árboles de sintaxis
- 5.5 Optimización de código
- 5.6 Generación de código estándar
- 5.7 Administración de la memoria durante la ejecución
- 5.8 Herramientas automáticas para generación de código
- 5.9 Programación de un compilador básico

### EXPERIENCIAS DE APRENDIZAJE

- Trabajar en equipo para la solución de problemas
- Integrar conocimientos previos de programación, estructura de datos y otros

### CRITERIOS DE EVALUACIÓN

Exámenes escritos	30%
Prácticas	30%
Proyecto final	40%
TOTAL	<u>100%</u>

### RECURSOS NECESARIOS

Recursos tecnológicos:

Visual C++, Java, Netbeans, Flex SDK.

### PRÁCTICAS SUGERIDAS

- Práctica de laboratorio para conocer el sistema operativo Linux y sus comandos para trabajar con el compilador Flex.
- Realizar análisis léxico en la herramienta Flex para el reconocimiento de: letras y números, operadores y símbolos, variables y cadenas, letras y números.
- Desarrollar un árbol sintáctico en herramienta computacional.
- Practica de laboratorio para realizar un análisis sintáctico básico para la validación de estructuras.
- Uso de herramientas automáticas para generar analizadores semánticos.
- Construcción de un pequeño compilador que genere código objeto.

### BIBLIOGRAFÍA BÁSICA

AHO, A., et al. (2008). Compiladores. Principios, técnicas y herramientas. México: Pearson.

ALFONSECA, M., et al. (2006). Compiladores e intérpretes. Teoría y práctica. México: Pearson Educación.

RUIZ, J. (2010). Compiladores: Teoría e implementación. México: Alfaomega.

#### **BIBLIOGRAFÍA COMPLEMENTARIA**

LEMONE, K. (1998). Fundamentos de compiladores. CECSA.

ALFONSECA, E., et al. (2007). Teoría de automatas y lenguajes formales. Madrid: McGraw Hill. ISBN: 9788448156374.

LINZ, P. (2001). An introduction to formal languages and automata. USA: Jones and Bartlett.

<http://www.hispafuentes.com/hf-doc/man/man1/flex++.1.html>

Página que contiene el manual de bison. Generador de analizadores sintácticos y semánticos.

<http://www.es.tldp.org/Manuales-LuCAS/BISON/bison-es-1.27.html>

Página tutorial de lex. Generador de analizador léxico y yacc generador de analizador sintáctico y semántico.

<http://mailweb.udlap.mx/~is111936/oto02/documentos/tutoriales/lex&yacc/>

Página que contiene todo lo relacionado con la construcción de compiladores e intérpretes.

<http://www.ucse.edu.ar/fma/compiladores/>