SILABO

1. INFORMACIÓN GENERAL

Asignatura	:	Lenguajes y Compiladores
Ciclo	:	8vo.
Área	:	Computación
Código	:	II0803
Condición	:	Obligatorio
Pre-requisito	:	Estructura de Datos y Algorítmica II
Horas Semanales	:	6 Horas
Teoría	:	3 Horas
Laboratorio	:	3 Horas
Créditos	:	3
Semestre Académico	:	2003-2

2. SUMILLA:

2.1 Naturaleza de la asignatura.

Es un curso teórico práctico.

2.2 Síntesis del Contenido

Se presentan los conceptos básicos, definiciones formales, técnicas utilizadas, las clases de compiladores, el contexto en el que se desarrollan, así como el tratamiento y recuperación de errores.
Se exponen y discuten las características de diseño, los fundamentos teóricos y los algoritmos que se utilizan en cada componente, así como las herramientas avanzadas de desarrollo de compiladores dando énfasis en las áreas de Teoría de Lenguajes, Lenguajes de Programación y Algoritmos.

3. OBJETIVO GENERALES:

- ✓ Adquirir conocimiento de la teoría de Lenguajes
- ✓ Conocer la arquitectura de los traductores
- ✓ Aplicación de los fundamentos teóricos en el desarrollo de software de base

Objetivos Específicos

- ✓ Capacitar al alumno en los conocimientos fundamentales de construcción de Compiladores para traducción de los Lenguajes de Programación de alto nivel.
- ✓ Realizar un proyecto de implantación de software de base.

4. PROGRAMA ANALITICO:

4.1 Aspectos Formales

Sistemas y Gramáticas Formales: alfabeto, lenguajes, símbolos terminales y no terminales. Jerarquía de lenguajes, Forma Normal de Backus (BNF).
Pre requisito: es necesario repasar los conceptos estudiados en el curso de Matemáticas Discretas: Lenguajes formales, gramáticas formales.
Bibliografía:

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 1
- *Teoría de autómatas y Lenguajes Formales.* Kelley. Cap. 1.
- *Matemática Discreta y lógica. Grassmann Cap. 10.*
- *Estructuras de Matemáticas Discretas para la computación. Kolman Cap. 10.*
- *The theory of parsing, translation, and compiling. Volume I: Parsing.* Aho. Cap. 2.

4.2 Arquitectura básica de un compilador

Lenguajes de programación. Compiladores: clases y estructura básica.
Diseño del compilador: descripción de las fases, análisis lexicográfico (scanner), análisis sintáctico (parser), análisis semántico, generación de código intermedio, generación de código objeto, tratamiento y recuperación de errores, administración de la tabla de símbolos.
Bibliografía:

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 1.
- *Compiladores, Principios, Técnicas y herramientas.* Aho. Cap. 1 y 2.
- *Lenguajes de Programación Diseño e Implementación.* Pratt. Cap. 1, 2 y 3.
- *The theory of parsing, translation, and compiling. Volume I: Parsing.* Aho. Cap. 1 y 3.

4.3 Análisis Lexicográfico

Funciones y reconocimiento de tokens. Especificación de los componentes lexicográficos. Gramáticas regulares. Tratamiento y recuperación de errores lexicográficos. Autómatas finitos y su implantación.
Pre requisito: es necesario repasar los conceptos estudiados en el curso de Matemáticas Discretas: Lenguajes y gramáticas formales, expresiones regulares, grafos, autómatas finitos, matriz de transición.
Bibliografía:

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 3.
- *Compiladores, Principios, Técnicas y herramientas.* Aho. Cap. 3.
- *Teoría de autómatas y Lenguajes Formales.* Kelley. Cap. 2 y 3.

4.4 Análisis Sintáctico

Métodos de análisis sintáctico: descendente y ascendente. Otros tipos de parser.
Bibliografía:

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 4.
- *Compiladores, Principios, Técnicas y herramientas.* Aho. Cap. 4.

4.5 Manejo de errores

Tipos de errores. Recuperación.
Bibliografía:

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 6.

4.6 Análisis sintáctico dirigido por tablas

Método LL(1): gramáticas LL(1), método de parser descendente, algoritmos para generar la tabla del parser.

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 4.
- *Compiladores, Principios, Técnicas y herramientas.* Aho. Cap. 7, 8, 9 y 10.
- Matemática Discreta y lógica. Grassmann Cap. 10.

4.7 Tablas de símbolos

Tabla de símbolos: uso, estructura, organización, primitivas de acceso.
Pre requisito: Repasar los temas de Estructuras de Datos, Organización de archivos métodos de búsqueda y clasificación de los cursos de Algorítmicas y Estructura de Datos.
Bibliografía:

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 3.
- *Compiladores, Principios, Técnicas y herramientas.* Aho. Cap. 7.

4.8 Análisis Semántico

Funciones de la Verificación semántica. Comprobación de tipos.
Bibliografía:

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 5.
- *Compiladores, Principios, Técnicas y herramientas.* Aho. Cap. 6.

4.9 Generación de Código

Código intermedio. Código Objeto. Asignación de memoria.

- *Compiladores Conceptos fundamentales*. Teufel. Cap. 7.
- *Compiladores, Principios, Técnicas y herramientas.* Aho. Cap. 7, 8, 9 y 10

- *The theory of parsing, translation, and compiling. Volume II: Compiling.* Aho. Cap. 11.

4.10 Sustentación de proyecto Informe 4 (2 horas)

En todas las exposiciones los grupos deberán exponer el avance de su proyecto, con la participación de todos los alumnos del bloque que podrán opinar y aportar sugerencias a los proyectos de sus compañeros. .

5. METODOLOGÍA:

- Las clases de la parte teórica se desarrollarán en aula presentando las principales técnicas aplicadas a cada una de las fases del diseño de compiladores. Además, se combinarán con lecturas obligatorias compuestas por artículos o capítulos de libros que se discutirán en clase, por lo que éstos deben ser leídos antes de clase.
- Las clases de práctica servirá para discutir los aspectos del desarrollo del proyecto
- La parte inicial del laboratorio tiene como objetivo dar a conocer la sintaxis y semántica del Lenguaje C++. Posteriormente se estudiará el uso de VCL para desarrollo de software de base.

6. EVALUACION:

6.1 Criterios

- ✓ Control de lecturas.
- ✓ Participación en aula.
- ✓ Evaluaciones rápidas (quiz).

6.2 Instrumentos

Concepto	Porcentaje	Compuesto	Responsable
Examen Parcial Teoría	**25%**	100% Examen Parcial.	Profesor de Teoría
Examen Final Teoría	**25%**	100% Examen Final.	Profesor de Teoría
Promedio de Prácticas de Teoría	**20%**	• 70% Proyecto • 30% Promedio de Quiz.	Profesor de Teoría
Promedio de Laboratorio	**30%**	• 70% Proyecto. • 30% Promedio de Quiz.	Profesor de Laboratorio

6.3 Proyecto

- ✓ Puntualidad en la presentación de informes y avances
- ✓ Informe Final
- ✓ Sustentación del Proyecto
- ✓ Ejecución del proyecto

7. BIBLIOGRAFÍA:

Lecturas básicas

- ✓ *Compiladores Conceptos fundamentales*.
  Teufel & Schmidt & Teufel. Addison-Wesley, 1996.
- ✓ *Compiladores, Principios, Técnicas y Herramientas*
  Alfred Aho & Ravi Sethi & Jeffrey Ullman. Addison-Wesley, 1990.
- ✓ *Teoría de Autómatas y Lenguajes Formales.*
  Kelley Dean,. Prentice Hall, 1995
- ✓ *The theory of parsing, translation, and compiling. Volume I: Parsing*
  Aho & J. Ullman, Prentice Hall, 1972.
- ✓ *The theory of parsing, translation, and compiling. Volume II: Compiling*
  Aho & J. Ullman, Prentice Hall, 1972.
- ✓ *Lenguajes de Programación Diseño e Implementación*
  Pratt Terrence & Zelkowitz Marvin. Prentice Hall 1998.
- ✓ *Matemática Discreta y lógica.*
  Grassmann Winfried Karl, Tremblay Jean-Paul. Prentice Hall 1997.
- ✓ *Estructuras de Matemáticas Discretas para la computación.*
  Kolman Bernard, Busby Robert C., Ross Sharon. Prentice Hall 1997.
- ✓ C++ From the Beginning.
  *Skansholm Jan. Addison Wesley*l 1997.
- ✓ Microsoft Visual C++ 6.0 Languaje Reference.
  Microsft Press 1998.
- ✓ Microsoft Visual C++ 6.0 MFC Library Reference. Part 1 & Part 2.
  Microsft Press 1998.
- ✓ Learn Microsoft Visual C++ 6.0 Now.
  Sphar Chuck. Microsft Press 1999.
- ✓ *C++ para Ingeniería y Ciencias.*
  Gary Bronson Thomson Editores 2000

Lecturas complementarias

- ✓ *Compiler Construction.*
  Nicklaus Wirth. Addison-Wesley, 1996.
- ✓ *Introduction to Automata Theory, Languages and Computation.*
  J. Hopcroft & J. Ullman. Addison-Wesley, 1979.
- ✓ *Construcción de Compiladores.*
  Gries, Paraninfo, 1971.
- ✓ *Compiler Design*.
  Wilhelm Reinhard & Maurer Dieter, Addison Wesley, 1995.
- ✓ *Introduction to Object Oriented Programming*
  Timothy Budd, Addison, Wesley,Second Edition 1997.
- ✓ *Construcción de software Orientado a Objetos*
  Bertrand Meyer. Prentice Hall, 1999

Programa Calendarizado de Lenguajes y Compiladores

Ciclo 2003- 2

Semana	Teoría	Laboratorio (3 horas)	Proyecto
1	Arquitectura de un compilador	Ejercicios y Programas con clases en C++	Definir el lenguaje
2	Arquitectura de un compilador. Aspectos Formales.	Programas con cadenas de caracteres	Definir el lenguaje
3	Aspectos Formales. Análisis Léxico	Programas de Análisis Léxico	*Entrega de Informe 1. Presentación del lenguaje*
4	Análisis Léxico.	Clases para Estructuras de datos	Modelo de arquitectura del proyecto
5	Análisis Léxico Análisis Sintáctico. Quiz 1	Clases para Estructuras de datos	Diseño e implementación del scanner
6	Análisis Sintáctico.	Clases para Estructuras de datos Quiz 1	Diseño e implementación del scanner
7	*Sustentación parcial del proyecto*	Clases para Estructuras de datos.	*Entrega de informe 2. Léxico*
8	**Parcial**		
9	Análisis Sintáctico	Tratamiento de excepciones	Diseño e implantación del parser
10	Manejo de Errores. Análisis Sintáctico dirigido por tablas Quiz 2	Tratamiento de excepciones	Diseño e implantación del parser
11	Análisis Sintáctico dirigido por tablas	Aspectos avanzados: desarrollo del sintáctico. Quiz 2	Diseño e implantación del parser. Manejo de Errores Entrega de Informe 3. Parser
12	Tabla de símbolos. Análisis Semántico.	Aspectos avanzados: desarrollo del semántico	Tabla de símbolos y análisis semántico
13	Análisis Semántico. Código Intermedio	Aspectos avanzados: desarrollo del generador	Generación de código intermedio
14	Generación de código.	Aspectos avanzados: desarrollo del intérprete. Práctica Calificada	Intérprete
15	*Sustentación del proyecto*	Sustentación del proyecto	Entrega de Informe Final
16	**Final**		