



DATOS DE IDENTIFICACIÓN DEL CURSO

DEPARTAMENTO:	CIENCIAS COMPUTACIONALES				
ACADEMIA A LA QUE PERTENECE:	SOFTWARE DE SISTEMAS				
NOMBRE DE LA MATERIA:	TALLER DE COMPILADORES				
CLAVE DE LA MATERIA:	CC318				
CARÁCTER DEL CURSO:	ESPECIALIZANTE				
TIPO DE CURSO:	CURSO-TALLER				
No. DE CRÉDITOS:	11				
No. DE HORAS TOTALES:	60	Presencial	60	No presencial	0
ANTECEDENTES:	TEORIA DE LA COMPUTACIÓN				
CONSECUENTES:					
CARRERAS EN QUE SE IMPARTE:	ING. COMPUTACIÓN				
FECHA DE ULTIMA REVISIÓN:	20 DE ENERO 2009				

PROPÓSITO GENERAL

Aprender a implementar los algoritmos aprendidos en la materia de teoría de compiladores. Comprender completamente como funcionan cada una de las fases del compilador. Tener la capacidad de generalizar los algoritmos implementados para resolver problemas de traducción de lenguajes, procesamiento de lenguajes y reconocimiento de patrones.

OBJETIVO TERMINAL

Comprender las técnicas utilizadas para el diseño de un compilador. Aprender cómo funcionan y como implementar cada una de las etapas del compilador: el análisis léxico, análisis sintáctico, análisis semántico y la generación de código. Además de saber cómo interactúan cada una de las etapas mencionadas.

CONOCIMIENTOS PREVIOS

Teoría de la computación, programación estructurada, estructura de datos, estructura de archivos, programación orientada a objetos, programación de sistemas.

HABILIDADES Y DESTREZAS A DESARROLLAR

El alumno será capaz de diseñar nuevos lenguajes, así como programas que los reconozcan. El alumno tendrá la adquirirá la habilidad suficiente para escribir programas en ensamblador, para después poder realizar la generación de manera automática. El alumno será capaz de trabajar con las diferentes estructuras de datos del compilador: símbolos, árbol sintáctico, tabla de símbolos.

Las técnicas aprendidas en el curso podrán ser aplicadas para resolver problemas de



concordancia de cadenas y reconocimiento de patrones, traducción de lenguajes. Tendrá la habilidad de diseñar e implementar gramáticas independientes de contexto y definiciones dirigidas por la sintaxis que le permitan resolver problemas de procesamiento de lenguajes.

ACTITUDES Y VALORES A FOMENTAR

Autoformación didáctica, respeto, puntualidad, disciplina, trabajo en equipo, ética profesional

METODOLOGÍA DE ENSEÑANZA APRENDIZAJE

Método	Método tradicional de exposición	Método Audiovisual	Aula Interactiva	Multimedia	Desarrollo de proyecto	Dinámicas	Estudio de casos	Otros (Especificar)
%	10%				75%		15%	



CONTENIDO TEMÁTICO

CONTENIDO TEMÁTICO		
MODULO 1. Analizador léxico		6 HRS
Implementar el analizador léxico del compilador		
1.1	Introducción	3 HRS
	Aprender como se implementan los autómatas finitos deterministas, como se pueden representar el analizador léxico y sus símbolos utilizando programación orientada a objetos.	
1.1.1	Implementa un analizador léxico para gramática de expresiones simple	
	Aprender como se implementan los autómatas finitos deterministas para realizar el análisis léxico	
1.1.2	Definición de la clase símbolo	
	Conocer la definición de la clase símbolo, es decir sus atributos y sus métodos.	
1.1.3	Definición de la clase que represente el analizador léxico	
	Conocer la definición de la clase analizador léxico, es decir sus atributos y sus métodos.	
1.2	Implementación del analizador léxico	3 HRS
	El alumno deberá implementar el analizador del análisis léxico del lenguaje de alto nivel indique.	
1.2.1	Definir los símbolos aceptados por el analizador léxico	
	Comprender como funciona el análisis léxico y como será utilizado en la fase del análisis sintáctico.	
1.2.2	Construir el autómata finito que acepte los símbolos léxicos del lenguaje	
	Comprender como funciona el análisis sintáctico, las tareas que debe realizar.	
1.2.3	Implementación del autómata finito	
	Comprender como funciona el análisis semántico, las tareas que debe realizar.	
1.2.4	Construir las clases para el símbolo léxico y el analizador léxico	
	Implementar la clase que realizara las tareas del analizador léxico.	



MODULO 2. Análisis Sintáctico

24 HRS

Implementar el analizador sintáctico del compilador, el cual deberá hacer el análisis del lenguaje y construir el árbol sintáctico.

2.1	Introducción		9 HRS
	OBJETIVO DEL TEMA		
2.1.1	Implementación de un analizador sintáctico para una gramática de expresiones simple		
	Determinar el objetivo del analizador léxico.		
2.1.2	Definición de la clase analizador sintáctico		
	Aprender cuales son los métodos y atributos de la clase que realizará el análisis sintáctico		
2.1.3	Construcción del árbol sintáctico durante la fase de análisis sintáctico		
	Aprender como se construye el árbol sintáctico de manera automática.		

2.2 Construcción del analizador sintáctico

15 HRS

Descripción de lenguajes utilizando los conceptos de teoría de la computación.

2.2.1	Definición del lenguaje aceptado		
	Conocer cual el lenguaje que deberá aceptar el analizador sintáctico.		
2.2.2	Implementación del analizador sintáctico		
	Definir completamente la clase del analizador sintáctico		
2.2.3	Implementar la generación automática de árboles sintácticos		
	Construir el árbol sintáctico que será utilizado en las siguientes fases.		

MODULO 3. Análisis semántico

6 HRS

Implementar el analizador semántico del compilador

3.1	Validación de tipos utilizando un árbol sintáctico		3 HRS
	Aprender como se realiza la validación de tipos utilizando un árbol sintáctico. El árbol sintáctico puede ser proporcionado por el profesor para que la implementación se enfoque en la validación de tipos.		
3.2	Tabla de símbolos		3 HRS
	Aprender como se implementa y se utiliza la tabla de símbolos.		



MODULO 4. Generación de código objeto		21 HRS
Implementar el generador de código utilizando el árbol sintáctico generado en las fases previas.		
4.1	Lenguaje ensamblador	6 HRS
	Aprender como se puede aumentar la capacidad expresiva una gramática, mediante el uso de atributos.	
4.1.1	Instalación del ensamblador	
	Comprender como funcionan los atributos sintetizados.	
4.1.2	Ensamblado / enlazado	
	Comprender como funcionan los atributos heredados.	
4.1.3	Escritura de programas	
	Comprender como funcionan los atributos heredados.	
4.2	Generación de código objeto	15 HRS
Aprender a construir arboles sintácticos utilizando los analizadores sintácticos descendentes y ascendentes.		
4.1.1	Implementación de la generación de código utilizando árboles sintácticos para una gramática de expresiones simple	
	Aprender como se implementa la generación de código objeto.	
4.1.2	Implementación de la generación de código utilizando árboles sintácticos para la gramática del lenguaje indicado	
	Implementar el código objeto del lenguaje de alto nivel	

CRITERIOS DE EVALUACIÓN

10% Practicas
90% Proyecto

BIBLIOGRAFÍA

BÁSICA				
TITULO	AUTOR	EDITORIAL	AÑO DE EDICIÓN	% DE COBERTURA DEL CURSO
Compiladores Principios técnicas y herramientas	Alfred V. Aho Monica S. Lam	Pearson	2008	90%



Segunda Edición	Ravi Sethi Jeffrey D. Ullman	Pearson		
Compiladores Principios técnicas y herramientas Primera Edición	Alfred V. Aho Ravi Sethi Jeffrey D. Ullman		1990	90%
COMPLEMENTARIA				
TITULO	AUTOR	EDITORIAL	AÑO DE EDICIÓN	% DE COBERTURA DEL CURSO
Construcción de compiladores	Kenneth C. Louden	Thomson	2004	70%
Teoría de la computación	J. Glenn Brookshear	Addison Wesley	1995	30%
Teoría de autómatas y lenguajes formales	Dean Kelly	Prentice Hall	1995	20%
Estructura de datos y Algoritmos	Aho, A. J. Hopcroft y J.	Addison Wesley		20%

REVISIÓN REALIZADA POR:

NOMBRE DEL PROFESOR	FIRMA
Dr. Carlos Alberto López Franco	

Vo.Bo. Presidente de Academia

Vo.Bo. Jefe del Departamento

miércoles, 28 de enero de 2009