



2016-B

ACADEMIA DE LENGUAJES INFORMÁTICOS					
I	NOMBRE DE LA MATERIA	SEMINARIO DE SOLUCION DE PROBLEMAS DE TRADUCTORES DE LENGUAJES II			
	TIPO DE ASIGNATURA	CURSO TEORICO-PRACTICO	CLAVE	17028	
II	CARRERA	LICENCIATURA EN INGENIERÍA EN COMPUTACION			
	ÁREA DE FORMACIÓN	ESPECIALIZANTE OBLIGATORIA			
III	PRERREQUISITOS	TRADUCTORES DE LENGUAJES I (17025) TEORÍA DE LA COMPUTACIÓN (15915)			
IV	CARGA GLOBAL TOTAL	TEORÍA	51	PRÁCTICA	17
V	VALOR EN CRÉDITOS	5			
FECHA DE CREACIÓN	JULIO 2016	FECHA DE MODIFICACIÓN	JULIO 2016	FECHA DE EVALUACIÓN	JULIO 2016

VI. OBJETIVO GENERAL

Desarrollar las capacidades necesarias para comprender las bases teóricas para la construcción e implementación de un compilador. Así como Comprender los algoritmos que se utilizan para resolver cada una de las fases del compilador. Al mismo tiempo el dominar completamente el proceso para convertir un lenguaje de alto nivel a un lenguaje de bajo nivel.

VII. CONTENIDO TEMÁTICO

- 1) Identifica y programa las diferentes fases de un compilador comprendiendo las diferentes fases que atraviesa un lenguaje de alto nivel a un lenguaje de bajo nivel.
- 2) Diseña y programa un analizador léxico de manera óptima a partir de autómatas de pila.
- 3) Analiza los elementos de un analizador léxico:

Objetivo: Diseñar de un autómata finito para la construcción de un lenguaje de programación propio. Crear y programa la tabla de tokens. Distinguir los errores léxicos. Definir las reglas de un lenguaje de programación propio. Construir un analizador léxico en Java o en C++.

- 3.1. Funciones del analizador léxico.
- 3.2. Componentes: léxicos, patrones y lexemas.
- 3.3. Determinación de los componentes, léxicos mediante autómatas finitos.
- 3.4. Tabla de tokens.
- 3.5. Errores léxicos.
- 3,6, Generadores de analizadores Léxicos.



- 4) **Crea un analizador sintáctico a partir de autómeta de estado finito definido que reconozca una gramática propia. (utilizando YACC o la librería JFlex de Java)**

Objetivo: *Identificar y crear una gramática formal. Determinar la sintaxis de una gramática utilizándola notación la BNF (Backus-Naur Form). Construir diagramas de sintaxis de un lenguaje propio. Eliminar la ambigüedad de una gramática. Distinguir los Errores sintácticos. Programar un analizador sintáctico (utilizar un generador de analizador sintáctico como YACC o la librería JFlex de Java).*

- 4.1. Gramáticas libres de contexto
- 4.2. Árboles de derivación.
- 4.3. Análisis sintáctico descendente
- 4.4. Análisis sintáctico ascendente
- 4.5. Diagramas de sintaxis
- 4.6. Eliminación de la ambigüedad.
- 4.7. Manejo de errores

- 5) **Diseña y programa arboles de expresiones dirigida por la sintaxis un analizador semántico para un meta-compilador.**

Objetivo: *Detectar los errores semánticos. Establecer las reglas para la conversión de tipos en las expresiones. Actualizar la tabla de conversión de símbolos y de direcciones. Programar un analizador semántico.*

- 5.1. Arboles de expresiones y acciones semánticas de un analizador sintáctico.
- 5.2. Tipos de analizadores semánticos
- 5.3. Expresiones de tipos.
- 5.4. Manejo de errores semánticos.
- 5.5. Estrategias para indagar sobre conocimientos previos.
- 5.6. Estrategias que promuevan la comprensión mediante la organización de información.

- 6) **Generar una máquina virtual que ejecute código intermedio a partir del código fuente de un lenguaje propio.**

Objetivo: *Traducir los tipos de notación para la conversión de expresiones: Infija, prefija y posfija. Representar expresiones mediante el código intermedio. Crear una máquina virtual para que ejecute el código intermedio de un lenguaje propio.*

- 6.1. Introducción
- 6.2. Notaciones



- 6.3. Representaciones de código intermedio
 - 6.4. Código P
 - 6.5. Triplos
 - 6.6. Cuádruplos
 - 6.7. Esquemas e generación
 - 6.8. Traducción de expresiones.
 - 6.9. Código ensamblador.
- 7) **Aplica diferentes tipos de optimización que permita eficiente el código intermedio propio.**

Objetivo: *Practicar las diferentes técnicas para la optimización del código intermedio generado. Analiza códigos intermedios para proponer mejoras.*

Tipos de optimizaciones:

- 7.1. Optimización de Mirilla.
- 7.2. A partir de expresiones locales
- 7.2. A partir de bucles
- 7.3. Optimización Global
- 7.4. Generación de código óptimo para expresiones.

VIII. MODALIDAD DEL PROCESO DE ENSEÑANZA-APRENDIZAJE

La modalidad propuesta para la impartición de este curso es:

1. **Presencial: 100% presencial**

Las actividades de enseñanza y aprendizaje se llevan a cabo en un aula o laboratorio.

IX. BIBLIOGRAFÍA BÁSICA

Ruiz Catalán, Jacinto. (2010). Compiladores: teoría e implementación. Primera edición. México: Alfaomega.

https://books.google.com.mx/books?id=6rJ9loiSPOoC&pg=PA1&hl=es&source=gbs_toc_r&cad=3#v=onepage&q&f=false

Alfonseca Moreno, Manuel. (2006). Compiladores e intérpretes: teoría y práctica. Madrid. Pearson

<http://es.slideshare.net/izyalyth/compiladores-e-interpretes-teoria-y-practica-31326505>

Cantú Treviño Thelma (2015) Teoría de autómatas un enfoque práctico. México. Pearson



X. CONOCIMIENTOS, APTITUDES, ACTITUDES, VALORES, CAPACIDADES Y HABILIDADES QUE EL ALUMNO DEBE ADQUIRIR

CONOCIMIENTOS:

APTITUDES:

Capacidad y disposición para el buen manejo de actividades de informática y habilidad para ejercer ciertas tareas minimizando tiempo y esfuerzo, logrando con esto las condiciones idóneas para realizar actividades dependiendo el área laboral.

ACTITUDES:

- Acepta y respeta las opiniones de los demás.
- Colabora con sus compañeros con la finalidad de mejorar el trabajo en equipo.
- Muestra interés al aprendizaje continuo y autogestivo.
- Valora la retroalimentación grupal.

VALORES:

Se pretende que el alumno al finalizar el curso, le permita manifestar su identidad en relación a sus nuevos conocimientos tanto en su trayecto escolar con su delación con el exterior.

CAPACIDADES:

El alumno tendrá la capacidad de poder resolver un problema, así como también mejorar los procesos en tiempo y forma para realizarlo dependiendo de las circunstancias en que se presente.

HABILIDADES:

El alumno tendrá la disposición para realizar tareas relacionadas con el área de desarrollo de Software, basándose en una adecuada percepción de los estímulos externos y una respuesta activa que redunde en una actuación eficaz, es decir, contara con el potencial para adquirir y manejar nuevos conocimientos y destrezas.

XI. CAMPO DE APLICACIÓN PROFESIONAL

Cualquier empresa que requiera del desarrollo de algoritmos para la solución de sus problemas, empresas destinadas al desarrollo e implementación de software de aplicaciones.



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE LA COSTA

DIVISIÓN DE INGENIERÍAS

DEPARTAMENTO DE CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

XII. EVALUACIÓN

Evaluación y acreditación

Área de conocimiento:

a) Reportes de investigación (20%)

Área de habilidades y destrezas:

a) Practicas individuales (35%)

b) Proyecto final (40%)

Área de actitud:

c) Participación 5%

(Podrá agregar cuantos elementos requiera en cada uno de los apartados de evaluación)

XIII. TIPO DE PRÁCTICAS

Realizara prácticas en la que podrá interpretar algoritmos computacionales y notaciones matemáticas (autómatas de estado finito y autómatas de pila), manejando la programación para la solución de aplicaciones, y aplicando las estructuras de datos en la solución de problemas, y manipular las operaciones básicas de los archivos.

XIV. MAESTROS QUE IMPARTEN LA MATERIA

Lic. Javier Joya Lomeli.

Código: 2830345

e-mail: javier_joya@hotmail.com

Formación. Licenciado en Sistemas Computacionales, Instituto de Ciencias y Estudios Superiores de Michoacán. Estudiante de la Maestría en Administración de Instituciones Educativas, becado del Instituto Tecnológico de Monterrey, Campus Guadalajara, Educación Virtual Via Internet (ITESM). Diplomado en Administración, Democracia y Desarrollo Municipal, UMSNH, Escuela de Economía, Edif.. "T", Ciudad Universitaria, Morelia, Mich., Diplomado en El Ciclo de Vida de los Proyectos de Inversión , OEA, UMSNH, Escuela de Economía, Edif.. "T", Ciudad Universitaria, Morelia, Mich.

Conocimientos y habilidades. Manejo de equipo de cómputo en mantenimiento preventivo, correctivo y adaptativo. Especialista en lenguajes de programación y programación orientada a objetos. Especialista en manejadores de bases de datos. Pleno conocimiento en el lenguaje de programación lógica y funcional. Programación y aplicaciones en ensamblados. Dominio de sistemas operativos basados en Windows y linux.



UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE LA COSTA

DIVISIÓN DE INGENIERÍAS

DEPARTAMENTO DE CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

XV. PROFESORES PARTICIPANTES

Lic. Javier Joya Lomeli.

CREACIÓN DEL CURSO:

Lic. Javier Joya Lomeli.

MODIFICACIÓN DEL CURSO:

Lic. Javier Joya Lomeli.

EVALUACIÓN DEL CURSO:

*Ing. Gustavo Viera Estrada
Presidente de la academia*

*Mtro. Anzoni herrera Martinez
Secretario Académico*

ING. GUSTAVO VIERA ESTRADA

*Presidente de Academia de lenguajes
informáticos*

DR. AURELIO ENRIQUE LÓPEZ BARRÓN

*Jefe del Departamento de Ciencias y
Tecnologías de la Información y
Comunicación*

DR. JOSE IGNACIO CHAVOYA GAMA

Director de la División de Ingenierías