

2408 – FUNDAMENTOS DE COMPILADORES



UASLP
Universidad Autónoma
de San Luis Potosí



**FACULTAD DE
INGENIERÍA**
Área de Ciencias
de la Computación

Clave de la materia: 2408
Clave Facultad: 2408
Clave U.A.S.L.P.: ----
Nivel del Plan de Estudios: I.C.: 7
Horas/Clase/Semana: 4
Horas/Práctica (y/o Laboratorio): 0
Prácticas complementarias: 0
Trabajo extra-clase Horas/Semana: 4
Carrera/Tipo de materia: I.C.: Obligatoria
No. de créditos aprobados: ----
Fecha última de Revisión Curricular: 30-junio-2023
Materia y clave de la materia requisito: 2206 -- Estructuras de Datos Avanzadas
Clave CACEI: CI
No. de créditos: 8
Horas totales/Semestre: 64

OBJETIVO DEL CURSO

Implementar diferentes técnicas de la fase de análisis, particularmente de las etapas de análisis léxico y sintáctico mediante la comprensión del proceso general que llevan a cabo los compiladores.

CONTENIDO TEMÁTICO

1. PROCESADORES DE LENGUAJE

Tiempo Estimado: 4 hrs.

Objetivo: Analizar y conocer el funcionamiento general de los procesadores de lenguaje.

- 1.1. Definición y breve historia
- 1.2. Conceptos básicos

2. ESTRUCTURA DE UN COMPILADOR

Tiempo Estimado: 4 hrs.

Objetivo: Entender y aplicar las ventajas de virtualizar recursos en un sistema de cómputo, para realizar ejercicios de virtualización de recursos como el procesador y la memoria en un sistema local.

- 2.1. Etapas del compilador
- 2.2. Tabla de símbolos

3. ANÁLISIS LÉXICO

Tiempo estimado: 24 hrs.

Objetivo: Analizar e implementar algoritmos del analizador léxico basados en operaciones con expresiones regulares y sus respectivos autómatas finitos para el reconocimiento de cadenas en lenguajes.

- 3.1. Conceptos básicos
- 3.2. Expresiones regulares
- 3.3. Autómatas finitos

4. ANÁLISIS SINTÁCTICO

Tiempo estimado: 16 hrs.

Objetivo: Analizar e implementar algoritmos del analizador sintáctico para determinar si existe un error de sintaxis o no.

- 4.1. Conceptos básicos
- 4.2. Análisis sintáctico descendente
- 4.3. Análisis sintáctico ascendente

METODOLOGÍA

Explicación de algoritmos y realización de problemas en todas las unidades (siguiendo el método de aprendizaje basado en problemas). A su vez el estudiante deberá realizar entregas de ejercicios de refuerzo de conocimiento (por parcial) para armar un portafolio de evidencias. Durante todo el curso se seguirá el método de aula invertida, en donde se espera que el estudiante tenga el compromiso de leer acerca del tema antes de

ciertas clases. También se seguirá el método basado en proyectos, en donde el estudiante deberá realizar un proyecto final en equipo de 2 personas, el cual realizarán durante el transcurso de la materia. El proyecto se divide en 8 entregas de manera modular para que al final se tenga un producto final terminado (se recomienda revisar los avances programados cada 2 semanas).

EVALUACIÓN

Se realizarán cuatro exámenes parciales de forma colegiada en las fechas establecidas por la Facultad, de acuerdo con el Reglamento de Exámenes. La calificación de los exámenes parciales estará compuesta por la evaluación del examen parcial y por otras actividades,

como: tareas, investigaciones, resolución de problemas, ejercicios, etc. La calificación del examen ordinario es el promedio de los cuatro parciales.

El proyecto es requisito para acreditar la materia.

BIBLIOGRAFÍA

Bibliografía Básica

Aho, A. V.; Lam, M.; Sethi, R.; Ullman, J. D.:
Compilers: Principles, Techniques, and Tools. 2a. Edición.
Addison-Wesley. 2013.

Thain, D. *Introduction to Compilers and Language Design*. Independently Published, 2020.

Vázquez J., Meloni B., Constable L. *Lenguajes formales y teoría de autómatas*. Alfaomega, 2016.

Moore, J., *Introduction to Compiler Design: An Object-Oriented Approach Using Java*. Softmoore Consulting, 2019.

Bibliografía Complementaria

Trigo, V. *Historia y evolución de los lenguajes de programación*. Autores científicos y académicos (ACTA) No. 34, 2004.

Wilhelm, R., Seidl, H., Hack, S., *Compiler Design: Syntactic and Semantic Analysis*, Springer, 2013