



NOMBRE DEL CURSO: Lenguajes Formales y de Programación

CÓDIGO:	796	CRÉDITOS:	3
ESCUELA:	Ciencias y Sistemas	ÁREA A LA QUE PERTENECE:	Ciencias de la Computación
PRE REQUISITOS:	770 – Introducción a la computación y programación 1 795 – Lógica de sistemas 960 – Matemática para la computación 1	POST REQUISITOS:	777 – Organización de lenguajes y compiladores 1 772 – Estructura de Datos
CATEGORÍA:	Obligatorio	SEMESTRE:	2do semestre 2018
CATEDRÁTICA:	ING. OTTO AMILCAR RODRIGUEZ ACOSTA	AUXILIAR:	Jorge Javier Veliz Florián
HORARIO DEL CURSO:	Martes 07:10 AM – 08:50 AM	HORARIO DEL LABORATORIO:	Lunes 09:00 AM – 10:50 AM

Descripción del curso:

Este laboratorio tiene como propósito poner en práctica los conocimientos teóricos adquiridos en el curso de Lenguajes Formales.

Durante el laboratorio, el estudiante aprenderá a implementar correctamente los elementos básicos de un compilador que están involucrados en las fases de análisis léxico, sintáctico y semántico.

El estudiante deberá diseñar gramáticas que reconozcan los lenguajes que se proponen en las prácticas y proyectos del laboratorio, y utilizar diversas técnicas y métodos de programación, para implementarlas.

Objetivo General:

Que el estudiante obtenga los conocimientos teóricos y prácticos que fundamentan el diseño de lenguajes de programación y compiladores.

Objetivos Específicos:

- Que el estudiante sea capaz de utilizar los conceptos teóricos para implementar un lenguaje formal.
- Que el estudiante sea capaz de desarrollar analizadores léxicos en base a autómatas finitos deterministas.
- Que el estudiante sea capaz obtener autómatas finitos deterministas a partir de expresiones regulares mediante el método de Thompson y la construcción de subconjuntos.
- Que el estudiante pueda obtener autómatas finitos deterministas a partir de expresiones regulares mediante el método del árbol.
- Que el estudiante sea capaz de desarrollar analizadores sintácticos en base a gramáticas independientes del contexto.

Metodología:

Se impartirán clases presenciales en las cuales se complementará el contenido teórico del curso, poniendo en práctica los conocimientos adquiridos mediante ejercicios y otras actividades prácticas.

Adicionalmente se realizarán proyectos, prácticas, tareas, exámenes cortos y hojas de trabajo con el objetivo de mejorar las habilidades de los estudiantes en el diseño y construcción de compiladores y evaluar los conocimientos adquiridos.

Dentro del desarrollo del laboratorio el alumno deberá tomar en cuenta lo siguiente:

- Copias parciales o totales en las tareas, investigaciones, etc. serán sancionadas.
- Copias en los proyectos y prácticas serán sancionadas y reportadas a la escuela de sistemas.
- Las tareas, investigaciones, prácticas, proyectos, etc., deben ser entregadas en la fecha indicada y con el formato establecido.

Evaluación:

La nota final del laboratorio, se distribuye en actividades de evaluación de la siguiente manera:

Tareas	5 puntos
Práctica 1	10 puntos
Primer proyecto	25 puntos
Práctica 2	10 puntos
Segundo proyecto	35 puntos
Examen final	15 puntos
Total	100 puntos

Observaciones:

- El laboratorio se debe de aprobar con una nota mínima de 61 puntos.
- Solo se calificarán exámenes, proyectos y demás actividades, a estudiantes asignados.
- Las prácticas y proyectos deben desarrollarse utilizando el lenguaje Visual Basic.

Contenido:

1. Conceptos generales
 - a. Definición de lenguaje formal
 - b. Definición de compilador
 - c. Fases del compilador
 - d. Definición de intérprete
 - e. Diferencias entre compiladores e intérpretes
2. Análisis Léxico
 - a. Token, lexema, patrón y ejemplos
 - b. Implementación de un analizador léxico
 - c. Tabla de símbolos
3. Jerarquía de Chomsky
 - a. Niveles
 - b. Restricciones
 - c. Ejemplos
4. Lenguajes regulares y gramáticas regulares
 - a. Definición
 - b. Diseño y construcción
 - c. Árboles de derivación
 - d. Ejemplos y aplicaciones
5. Expresiones Regulares
 - a. Definición y Propiedades
 - b. Diseño y construcción
 - c. Relación Gramáticas Regulares – Expresiones Regulares
6. Autómatas Finitos
 - a. Definición
 - b. Tabla de transición
 - c. Autómatas Finitos No Deterministas
 - d. Autómatas Finitos Deterministas
 - e. Implementación de Autómatas Finitos Deterministas
7. Método del Árbol
 - a. Construcción del árbol
 - b. Cálculo de primeros, últimos y siguientes.
 - c. Construcción del autómata finito determinista
 - d. Resolución de ejercicios aplicando el método
8. Método de Thompson y subconjuntos
 - a. Reglas
 - b. Resolución de ejercicios aplicando el método
9. Análisis Sintáctico
 - a. Funcionamiento de un analizador sintáctico
 - b. Implementación de analizadores sintácticos
 - c. Análisis sintáctico ascendente
 - d. Análisis sintáctico descendente
10. Gramáticas independientes del contexto
 - a. Árboles de derivación
 - b. Ambigüedad y recursividad

- c. Factorización por la izquierda
 - d. Eliminación de la recursividad por la izquierda
 - e. Implementación de autómatas de pila
 - f. Analizador sintáctico descendente
11. Analizador sintáctico descendente LL(1)
- a. Primeros y siguientes
 - b. Construcción de tabla de análisis sintáctico

CRONOGRAMA DE ACTIVIDADES	
FECHA DE ENTREGA	ACTIVIDAD
01/08/18 - 8/08/18	Tarea 1: Realizar ejercicios con expresiones regulares y gramáticas regulares
8/08/18 - 15/08/18	Tarea 2: Ejercicio con Autómatas tanto deterministas y no deterministas.
22/08/18 – 31/08/18	Tarea 3: Ejercicios para resolver con método del Árbol.
12/09/18 – 19/09/18	Tarea 4: Ejercicios para resolver con método de Thompson.
10/10/18 – 17/10/18	Tarea 5: Ejercicios para resolver con Autómata de Pila.
31/10/18	Examen final: Hecho mediante la plataforma de Socrative
29/07/18 – 10/08/18	Primer Practica: Expresiones Regulares.
12/08/18 – 07/09/18	Primer Proyecto: Analizador Léxico.
09/09/18 – 21/09/18	Segunda Practica: Análisis Léxico e introducción Sintáctico.
30/09/18 – 26/10/18	Segundo Proyecto: Análisis Léxico y Sintáctico.

Bibliografía:

- Aho, Alfred V., Sethi y Ullman. Compiladores: principios, técnicas y herramientas. Addison-Wesley.
- □
- Brookshear, J. Glenn. Teoría de la Computación - Lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana. □ □
- Andrew W. Appel. Modern Compiler Implementation in Java. Second Edition. Cambridge University Press.
- □
- Hopcroft, John y Ullman, Jeffrey. Introducción a la Teoría de Autómatas, Lenguajes y Computación. □
- Kenneth Loudon. Construcción de compiladores □