

NOMBRE DEL CURSO: Laboratorio de Lenguajes Formales y de Programación

CODIGO:	796	CREDITOS:	3
ESCUELA:	Ciencias y Sistemas	AREA A LA QUE PERTENECE:	Ciencias de la computación
PRE REQUISITOS:	770 – Introducción a la Programación 1 795 – Lógica de sistemas 960 – Matemática de cómputo 1	POST REQUISITO:	777 Organización de Lenguajes y Compiladores 1 772 Estructuras de Datos
CATEGORIA:	Obligatorio	SEMESTRE:	1^{er}. 2018
CATEDRÁTICO (A):	Ing. Zulma Karina Aguirre Ordoñez	AUXILIAR:	José Alberto Velásquez Orozco
EDIFICIO:	T-3	SECCIÓN:	B-
SALON DEL CURSO:	211	SALON DEL LABORATORIO:	T3-215
HORAS POR SEMANA DEL CURSO:	2	HORAS POR SEMANA DEL LABORATORIO:	2
DÍAS QUE SE IMPARTE EL CURSO:	Martes	DIAS QUE SE IMPARTE EL LABORATORIO:	Miércoles
HORARIO DEL CURSO:	07:00 – 08:40 HRS	HORARIO DEL LABORATORIO:	07:00 – 08:40 HRS

DESCRIPCIÓN DEL CURSO:

El laboratorio tiene como propósito introducir al estudiante de ciencias de la computación al estudio, análisis, comprensión e implementación de lenguajes de programación bajo una estructura genérica que contribuya al desarrollo de un compilador básico y funcional; abarcando las fases de análisis léxico, análisis sintáctico e introducción al análisis semántico.

Comprende también mostrar métodos que faciliten la creación de analizadores léxicos por medio de algoritmos programados.

OBJETIVOS:

Objetivo General

- Introducir al estudiante al conocimiento y desarrollo de las funciones básicas de los compiladores.

Objetivos Específicos

- Aplicar los conocimientos adquiridos en clase para implementar analizadores léxicos.
- Implementar un analizador sintáctico utilizando las técnicas vistas en clase.

METODOLOGIA:

- Se imparten clases presenciales con material de apoyo que es proporcionado al estudiante al finalizar la clase.
- En el transcurso del semestre se realizarán tareas y exámenes cortos para evaluar los conocimientos adquiridos.
- Se realizan prácticas y proyectos en donde se ponga en práctica y se puedan evaluar los conceptos adquiridos en el curso, tomando en cuenta que pueden incluirse temas de cursos pre requisito.
- Se impartirán clases prácticas donde se resuelvan problemas relacionados con el tema a desarrollar.

REQUISITOS:

- El laboratorio se debe aprobar con nota mínima de 61 puntos.
- Es obligatorio aprobar el laboratorio para tener derecho a examen final.
- Solo se calificarán exámenes y proyectos de estudiantes asignados en el curso. NO se agregan estudiantes a actas
- Será necesario contar con un 80% de asistencia.
- En este curso, no se pasan notas de semestres anteriores, no se guardan notas para semestres posteriores, y no se aceptan estudiantes con problemas de prerrequisitos.
- ***Las prácticas y proyectos deben desarrollarse utilizando lenguaje C#***
- 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.

EVALUACION:

Aspecto	Valor
Tareas, cortos y asistencia	10 pts.
Practicas	
<u>Practica 1</u>	15 pts.
Publicación: 14/02/18	
Entrega: 07/03/18	
Proyectos	
<u>Proyecto 1</u>	30 pts.
Publicación: 07/03/18	
Entrega: 04/04/18	
<u>Proyecto 2</u>	35 pts.
Publicación: 04/04/18	
Entrega: 02/05/18	
Examen final	<u>10pts</u>
	Total 100 pts.

*Para tener derecho a examen final se requiere una asistencia mayor o igual a 80%.

CONTENIDO

1. Lenguajes

- 1.1 Lenguajes Naturales
- 1.2 Características de lenguajes naturales
- 1.3 Lenguajes Formales
- 1.4 Características de lenguajes formales
- 1.5 Lenguajes de programación

2. Introducción a compiladores

- 2.1 ¿Qué es un compilador?
- 2.2 Ejemplos de compiladores
- 2.3 Diferencia entre compilador e interprete
- 2.4 Partes del compilador

3. Análisis léxico

3.1 Análisis lexicográfico

3.2 Token

3.3 Lexema

3.4 Patrón

4. Jerarquía de Chomsky

4.1 Tipos de gramáticas

4.2 Restricciones

4.3 Ejemplos

5. Lenguajes regulares

5.0 Lenguajes regulares

5.1 Expresiones regulares

5.2 Ejemplos

6. Autómatas finitos

6.1. Definición

6.2. Tabla de transiciones

6.3. Autómatas Finitos Deterministas (AFD)

6.4. Autómatas Finitos No Deterministas (AFND)

6.5. Implementación de AFD's

7. Método de Thompson

7.1. Nomenclatura

7.2. Cerraduras y subconjuntos

7.3. Transiciones

8. Método del árbol

8.1. Construcción de arboles

8.2. Calculo de primeros, últimos y siguientes

8.3. Subconjuntos y transiciones

8.4. Implementación del método del árbol

9. Implementación de un analizador léxico

9.1 Tabla de símbolos

9.2 Manejo de errores

10. Lenguajes libres de contexto

10.1 Lenguajes Independientes del contexto

10.2 Gramáticas Tipo 2

10.3 Diseño de gramáticas independientes del contexto

10.3.1 Ejemplos y ejercicios de gramáticas libres de contexto

10.4 Recursividad

10.4.1 Por la izquierda

10.4.2 Por la derecha

10.5 Gramáticas ambiguas

10.6 Autómatas de Pila
10.7 Parser recursivo descendente
10.8 Ejemplos y ejercicios

BIBLIOGRAFÍA:

- Aho, Alfred V., Sethi y Ullman. Compiladores: principios, técnicas y herramientas. Addison-Wesley.
- John E Hopcroft. introducción a la Teoría de Autómatas, Lenguajes y computación.
- Appleby, Doris y Vandekopple, Julius J. Lenguajes de Programación - Paradigma y práctica. Editorial McGraw Hill.