

**NOMBRE DEL CURSO: Laboratorio de Lenguajes Formales y de Programación**

<b>CODIGO:</b>	<b>796</b>	<b>CREDITOS:</b>	<b>3</b>
<b>ESCUELA:</b>	<b>Ciencias y Sistemas</b>	<b>AREA A LA QUE PERTENECE:</b>	<b>Ciencias de la computación</b>
<b>PRE REQUISITOS:</b>	<b>770 – Introducción a la Programación 1 795 – Lógica de sistemas 960 – Matemática de cómputo 1</b>	<b>POST REQUISITO:</b>	<b>777 Organización de Lenguajes y Compiladores 1 772 Estructuras de Datos</b>
<b>CATEGORIA:</b>	<b>Obligatorio</b>	<b>SEMESTRE:</b>	<b>1er. 2019</b>
<b>CATEDRÁTICO (A):</b>	<b>Inga. Damaris Campos</b>	<b>AUXILIAR:</b>	<b>Aylin Zulema Aroche Arbizú</b>
<b>EDIFICIO:</b>	<b>T-3</b>	<b>SECCIÓN:</b>	<b>A-</b>
<b>SALON DEL CURSO:</b>	<b>215</b>	<b>SALON DEL LABORATORIO:</b>	<b>213</b>
<b>HORAS POR SEMANA DEL CURSO:</b>	<b>2</b>	<b>HORAS POR SEMANA DEL LABORATORIO:</b>	<b>2</b>
<b>DÍAS QUE SE IMPARTE EL CURSO:</b>	<b>Martes</b>	<b>DÍAS QUE SE IMPARTE EL LABORATORIO:</b>	<b>Martes</b>
<b>HORARIO DEL CURSO:</b>	<b>07:00 – 08:40 HRS</b>	<b>HORARIO DEL LABORATORIO:</b>	<b>09:00 – 10:40 HRS</b>

**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, y los ponga en práctica en la construcción de las primeras fases de esté.

**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 en acta de notas.
- 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# (Visual Studio 2015/2017)**
- Copias parciales o totales en las tareas, investigaciones, etc. serán sancionadas con una nota de cero.
- Copias en los proyectos y prácticas serán sancionadas con una nota de cero y reportadas a la Escuela de Sistemas.
- Las tareas, investigaciones, prácticas, proyectos, deben ser entregadas en la fecha indicada y con el formato establecido.

**EVALUACION:**

<b>Aspecto</b>	<b>Valor</b>
Tareas, cortos y asistencia	10 pts.
Prácticas	
<u>Práctica 1</u>	15 pts.
Publicación: 05/02/19	
Entrega: 23/02/19	
Proyectos	
<u>Proyecto 1</u>	30 pts.
Publicación: 25/02/19	
Entrega: 25/03/19	
<u>Proyecto 2</u>	35 pts.
Publicación: 26/03/19	
Entrega: 26/04/19	
Examen final	<u>10 pts.</u>
	<b>Total 100 pts.</b>

\*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, Patrón, Lexema
- 3.3 Errores léxicos

### 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 del árbol

- 7.1. Construcción de arboles
- 7.2. Calculo de primeros, últimos y siguientes
- 7.3. Subconjuntos y transiciones
- 7.4. Implementación del método del árbol

### 8. Método de Thompson

- 8.1. Nomenclatura
- 8.2. Cerraduras y subconjuntos
- 8.3. Transiciones

### 9. Implementación de un analizador léxico

- 9.1 Tabla de símbolos
- 9.2 Programación de un autómata finito
- 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
- 10.4 Ejemplos y ejercicios de gramáticas libres de contexto
- 10.5 Recursividad
  - 10.5.1 Por la izquierda
  - 10.5.2 Por la derecha
- 10.6 Gramáticas ambiguas
- 10.7 Autómatas de Pila
- 10.8 Parser recursivo descendente
- 10.9 Ejemplos y ejercicios

## 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.
- John E Hopcroft. introducción a la Teoría de Autómatas, Lenguajes y computación.