

NOMBRE DEL CURSO: Laboratorio de Lenguajes Formales y de Programación

CÓDIGO:	796	CRÉDITOS:	3
ESCUELA:	Ciencias y Sistemas	ÁREA A LA QUE PERTENECE:	Ciencias de la computación
PRERREQUISITOS:	770 – Introducción a la Programación 1 795 – Lógica de sistemas 960 – Matemática de cómputo 1	POST REQUISITOS:	777 Organización de Lenguajes y Compiladores 1 772 Estructuras de Datos
CATEGORÍA:	Obligatorio	SEMESTRE:	1er Semestre. 2020
CATEDRÁTICO:	Ing. David Estuardo Morales Ajcot	AUXILIAR:	José Manuel Véliz Sandoval
EDIFICIO:	T-3	SECCIÓN:	B+
SALÓN DEL CURSO:	313	SALÓN DEL LABORATORIO:	T-3 India 1 T-3 014
HORAS POR SEMANA DEL CURSO:	2	HORAS POR SEMANA DEL LABORATORIO:	2
DÍAS QUE SE IMPARTE EL CURSO:	Martes	DÍAS QUE SE IMPARTE EL LABORATORIO:	Lunes
HORARIO DEL CURSO:	07:00 – 08:40 HRS	HORARIO DEL LABORATORIO:	07:10 – 8:50 AM 9:00 – 10:40 AM

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 prerrequisito.
- 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.
- 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.
- 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:

Aspecto:	Valor:
4 Tareas	5 pts. – 1.25 pts. c/u
3 Cortos	15 pts. – 5 pts. c/u
Practicas	
<u>Práctica 1:</u>	15 pts.
Publicación: 08/02/2020	
Entrega: 22/02/2020	
Proyectos	
<u>Proyecto 1:</u>	20 pts.
Publicación: 23/02/2020	
Entrega: 24/03/2020	
<u>Proyecto 2:</u>	35 pts.
Publicación: 03/04/2020	
Entrega: 04/05/2020	
Examen Final	10 pts.
Total	100 pts.

CALIFICACIÓN DE PRÁCTICAS Y PROYECTOS

- La calificación de las prácticas y proyectos se realizarán presencialmente y desde los archivos ejecutables entregados.
- No se puede agregar o quitar algún símbolo en los archivos de entrada. La práctica o el proyecto deberá funcionar con los archivos que sean proporcionados por el auxiliar para la calificación.
- Existirán horarios para la calificación, por lo cual el estudiante deberá de elegir el horario que mejor le convenga.
- Anomalías o copias detectadas de entregables tendrán de manera automática una nota de 0 puntos y los involucrados serán reportados a la Escuela de Ingeniería en Ciencias y Sistemas, para que se apliquen las sanciones correspondientes.

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 un compilador y un intérprete.
2.4	Partes de un 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	
5.1	Lenguajes regulares.
5.2	Expresiones regulares.
5.3	Ejemplos.
6. Autómatas finitos	
6.1	Definición.
6.2	Tabla de transiciones.
6.3	Autómatas Finitos Deterministas (AFD).
6.4	Autómatas Finitos No Deterministas (AFN).
6.5	Implementación de AFD's.

7. Método del árbol	
7.1.	Construcción de árboles.
7.2.	Cálculo 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. AddisonWesley Iberoamericana.
- John E Hopcroft. introducción a la Teoría de Autómatas, Lenguajes y computación.