

NOMBRE DEL CURSO: Laboratorio de Lenguajes Formales y de Programación

CÓDIGO:	796	CRÉDITOS:	3
ESCUELA:	Ciencias y Sistemas	AREA A LA QUE PERTENECE:	Ciencias de la Computación
PRE REQUISITOS:	770 – Introducción a la Programación y Computación 1 795 – Lógica de Sistemas 960 – Matemática de Cómputo 1	POST REQUISITOS:	777 - Organización de Lenguajes y Compiladores 1 772 - Estructuras de Datos
CATEGORIA:	Obligatorio	SEMESTRE:	Primero 2023
CATEDRÁTICO (A):	Ing. Otto Amilcar Rodriguez Acosta	AUXILIAR:	Erick Daniel Antillón Chinchilla
EDIFICIO:	MEET	SECCIÓN:	A+
SALON DEL CURSO:	SALÓN 22	SALÓN DEL LABORATORIO:	SALÓN 16
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:	Sábado
HORARIO DEL CURSO:	07:10 – 08:50 HRS	HORARIO DEL LABORATORIO:	10:30– 12:10 HRS

DESCRIPCIÓN DEL CURSO:

El laboratorio tiene como propósito introducir al estudiante de ciencias de la computación al estudio, análisis y comprensión de lenguajes de programación bajo una estructura genérica que contribuya al desarrollo de las capacidades de manejo y diseño de gramáticas del estudiante; abarcando conocimientos de modelos matemáticos que las resuelven y de lenguajes reales conocidos donde se pueden implementar.

OBJETIVOS:

Objetivo General

- Introducir al estudiante al conocimiento y desarrollo de los conceptos teóricos y matemáticos necesarios que fundamentan los lenguajes formales y de programación; mediante la clasificación de gramáticas, y el diseño de lenguajes mediante autómatas, expresiones y gramáticas.

Objetivos Específicos

- Diseñar gramáticas que representen lenguajes específicos.
- Aplicar los conocimientos adquiridos en clase para implementar soluciones en el lenguaje Python.

METODOLOGÍA:

- Se imparten clases con material de apoyo que es proporcionado al estudiante al finalizar la clase.
- 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.

RESTRICCIONES:

- Copias parciales o totales en la elaboración de tareas, hojas de trabajo, investigaciones, etc. serán sancionadas con nota de cero puntos.
- Copias en prácticas y proyectos serán sancionadas con una nota de cero puntos y reportadas a la Escuela de Sistemas.
- Las tareas, prácticas, proyectos y cualquier otra actividad deben ser entregados bajo un formato establecido en la fecha y hora indicadas.

OBSERVACIONES:

- Solo se calificarán exámenes, prácticas, proyectos y demás actividades de estudiantes asignados en el curso. NO se agregan estudiantes en acta de notas.
- NO se aceptan estudiantes con problemas de prerrequisitos; por ende, NO se pasan notas de semestres anteriores y no se guardan notas para semestres posteriores.
- **Es obligatorio aprobar el laboratorio para tener derecho a examen final de la clase magistral.**
- **Se debe contar con un 80% de asistencia para aprobar el laboratorio.**
- **El laboratorio debe ser aprobado con una nota mínima de 61 puntos.**
- **Las prácticas y proyectos deben desarrollarse utilizando el lenguaje Python.**

EVALUACION: El laboratorio se evalúa sobre una nota de 100 puntos teniendo 61 puntos como nota mínima de promoción. El detalle de la ponderación es el siguiente:

Aspecto	Valor
- Tareas	4 pts.
- Cortos	6 pts.
- Práctica	15 pts.
Publicación: 28/07/2023	
Entrega: 11/08/2023	
- Proyectos	
<u>Proyecto 1</u>	30 pts.
Publicación: 19/08/2023	
Entrega: 21/09/2022	
<u>Proyecto 2</u>	35 pts.
Publicación: 22/09/2022	
Entrega: 26/09/2022	
- Examen final	10 pts.
	Total 100 pts.

CONTENIDO:

1. Unidad 1: Python

- 1.1. Introducción al lenguaje
 - 1.1.1. Historia
 - 1.1.2. Aspectos básicos
- 1.2. Buenas practicas
- 1.3. Librerías
- 1.4. Clases, Métodos y Funciones
- 1.5. Arreglos
- 1.6. Diccionarios
- 1.7. Iteraciones
 - 1.7.1. Ciclos For
 - 1.7.2. Ciclos While
- 1.8. Archivos
 - 1.8.1. Lectura
 - 1.8.2. Escritura

2. Unidad 2: Lenguajes

- 2.1. Lenguajes
 - 2.1.1. Lenguaje Natural
 - 2.1.2. Lenguajes Formales
 - 2.1.3. Lenguajes de Programación
- 2.2. Evolución de los Lenguajes de Programación
 - 2.2.1. Paradigmas
 - 2.2.2. Generaciones de los lenguajes de programación
- 2.3. Procesadores de Lenguaje
 - 2.3.1. Intérprete
 - 2.3.2. Compilador
 - 2.3.2.1. Estructura de un compilador
 - 2.3.2.2. Herramientas
 - 2.3.3. Diferencias y Ejemplos
- 2.4. Jerarquía de Chomsky
 - 2.4.1. Clasificación de gramáticas
 - 2.4.2. Definiciones

3. Unidad 3: Análisis Léxico

- 3.1. Definición y Función del analizador Léxico
 - 3.1.1. Patrones, Tokens y Lexemas
 - 3.1.2. Errores Léxicos
- 3.2. Operaciones entre lenguajes
- 3.3. Expresiones Regulares
- 3.4. Diagrama de transición de estados
 - 3.4.1. Tablas de transición
- 3.5. Autómatas Finitos
 - 3.5.1. Autómatas finitos no deterministas (AFN)
 - 3.5.2. Autómatas finitos deterministas (AFD)

3.6. Conversión AFN a AFD

- 3.6.1. Método del Árbol
 - 3.6.1.1. Anulables
 - 3.6.1.2. Cálculo de *First*, *Last* y *Next*
- 3.6.2. Optimización de estados

4. Unidad 4: Análisis Sintáctico

- 4.1. Función del analizador sintáctico
- 4.2. Lenguajes libres de contexto
- 4.3. Gramáticas Tipo 2
 - 4.3.1. Árboles de derivación
 - 4.3.2. Ambigüedad
 - 4.3.3. Recursividad
- 4.4. Autómatas de Pila
 - 4.4.1. Procesamiento
 - 4.4.2. Tipos de aceptación
 - 4.4.2.1. Estado Final
 - 4.4.2.2. Pila vacía
 - 4.4.3. Teorema 2.2 (Autómata de Pila desde Gramática tipo 2)

BIBLIOGRAFÍA:

- Aho, A., Lam, M., Sethi, R. & Ullman, J. (2007). ***Compilers: Principles, Techniques & Tools*** (2nd ed.). Pearson.

Adicionales:

- Linz, P. (2017). ***An Introduction to Formal Languages and Automata*** (6th ed.). Jones & Bartlett Learning.
- Hopcroft, J., Motwani, R. & Ullman, J. (2007). ***Introduction Automata, Theory, Languages and Computation***. (3rd ed.). Pearson.
- Loudon, K. (2004). ***Lenguajes de programación: principios y práctica***. (2nd ed.). Cengage Learning.