

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

<b>CÓDIGO:</b>	796	<b>CRÉDITOS:</b>	3
<b>ESCUELA:</b>	Ciencias y Sistemas	<b>AREA A LA QUE PERTENECE:</b>	Ciencias de la Computación
<b>PRE REQUISITOS:</b>	770 – Introducción a la Programación y Computación 1 795 – Lógica de Sistemas 960 – Matemática de Cómputo 1	<b>POST REQUISITOS:</b>	777 - Organización de Lenguajes y Compiladores 1 772 - Estructuras de Datos
<b>CATEGORIA:</b>	Obligatorio	<b>SEMESTRE:</b>	Segundo 2021
<b>CATEDRÁTICO (A):</b>	Inga. Zulma Karina Aguirre	<b>AUXILIAR:</b>	Douglas Omar Arreola Martínez
<b>EDIFICIO:</b>	MEET	<b>SECCIÓN:</b>	B-
<b>SALON DEL CURSO:</b>	SALÓN 8	<b>SALÓN DEL LABORATORIO:</b>	SALÓN 26
<b>HORAS POR SEMANA DEL CURSO:</b>	2	<b>HORAS POR SEMANA DEL LABORATORIO:</b>	2
<b>DÍAS QUE SE IMPARTE EL CURSO:</b>	Martes	<b>DIAS QUE SE IMPARTE EL LABORATORIO:</b>	Sábado
<b>HORARIO DEL CURSO:</b>	07:10 – 08:50 HRS	<b>HORARIO DEL LABORATORIO:</b>	13:50 – 15:30 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:

<b>Aspecto</b>	<b>Valor</b>
- Tareas y cortos	
4 tareas	(1 punto cada una) 04 pts.
3 cortos	(1 punto cada uno) 03 pts.
- Hojas de trabajo	
2 hojas de trabajo	(1.5 puntos cada una) 03 pts.
- Práctica	15 pts.
Publicación:	29/07/2021
Entrega:	19/08/2021
- Proyectos	
<u>Proyecto 1</u>	30 pts.
Publicación:	20/08/2021
Entrega:	23/09/2021
<u>Proyecto 2</u>	35 pts.
Publicación:	24/09/2021
Entrega:	28/10/2021
- Examen final	10 pts.
	<b>Total 100 pts.</b>

## CONTENIDO:

### 1. Unidad 1: Lenguajes

- 1.1. Lenguajes
  - 1.1.1. Lenguaje Natural
  - 1.1.2. Lenguajes Formales
  - 1.1.3. Lenguajes de Programación
- 1.2. Evolución de los Lenguajes de Programación
  - 1.2.1. Paradigmas (imperativos / declarativos)
  - 1.2.2. Generaciones de los lenguajes de programación
- 1.3. Procesadores de Lenguaje
  - 1.3.1. Intérprete
  - 1.3.2. Compilador
    - 1.3.2.1. Estructura de un compilador
    - 1.3.2.2. Herramientas
  - 1.3.3. Diferencias y Ejemplos
- 1.4. Jerarquía de Chomsky
  - 1.4.1. Clasificación de gramáticas
  - 1.4.2. Definiciones

### 2. Unidad 2: Análisis Léxico

- 2.1. Definición y Función del analizador Léxico
  - 2.1.1. Patrones, Tokens y Lexemas
  - 2.1.2. Errores Léxicos
- 2.2. Operaciones entre lenguajes
- 2.3. Expresiones Regulares
- 2.4. Diagrama de transición de estados
  - 2.4.1. Tablas de transición
- 2.5. Autómatas Finitos
  - 2.5.1. Autómatas finitos no deterministas (AFN)
  - 2.5.2. Autómatas finitos deterministas (AFD)

### 2.6. Conversión AFN a AFD

- 2.6.1. Método Thomson
  - 2.6.1.1. Cerraduras
  - 2.6.1.2. Transiciones
  - 2.6.1.3. Sub conjuntos
- 2.6.2. Método del Árbol
  - 2.6.2.1. Cálculo de *First*, *Last* y *Next*
- 2.6.3. Optimización de estados

### 3. Unidad 3: Análisis Sintáctico

- 3.1. Función del analizador sintáctico
- 3.2. Lenguajes libres de contexto
- 3.3. Gramáticas Tipo 2
  - 3.3.1. Árboles de derivación
  - 3.3.2. Ambigüedad
  - 3.3.3. Recursividad
- 3.4. Autómatas de Pila
  - 3.4.1. Notación gráfica
  - 3.4.2. Procesamiento
  - 3.4.3. Tipos de aceptación
    - 3.4.3.1. Estado Final
    - 3.4.3.2. Pila vacía
  - 3.4.4. Teorema 2.2 (AP desde Gramática tipo 2)

## BIBLIOGRAFÍA:

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

### Adicionales:

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