

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 2024
CATEDRÁTICO (A):	Inga. Damaris Campos González	AUXILIAR:	Enrique Alejandro Pinula Quiñonez
EDIFICIO:	T-7	SECCIÓN:	A-
SALON DEL CURSO:	102	SALÓN DEL LABORATORIO:	Google Meet
HORAS POR SEMANA DEL CURSO:	1 hora con 40 minutos	HORAS POR SEMANA DEL LABORATORIO:	1 hora con 40 minutos
DÍAS QUE SE IMPARTE EL CURSO:	Martes	DIAS QUE SE IMPARTE EL LABORATORIO:	Viernes
HORARIO DEL CURSO:	07:10 – 08:50 HRS	HORARIO DEL LABORATORIO:	10:40 – 12:20 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.**
- **Obligatoria la entrega del segundo proyecto con un mínimo del 50%, de lo contrario se reprueba.**

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, cortos y hojas de trabajo	10 pts.
- Práctica	15 pts.
Publicación: 05/02/2024	
Entrega: 20/02/2024	
- Proyectos	
<u>Proyecto 1</u>	25 pts.
Publicación: 21/02/2024	
Entrega: 21/03/2024	
<u>Proyecto 2</u>	40 pts.
Publicación: 22/03/2024	
Entrega: 25/04/2024	
- Examen final	10 pts.
	Total 100 pts.

Tabla de contenido

1. Lenguajes Formales	1.2 Clasificación de los lenguajes de programación por niveles. 1.2.1 Lenguaje máquina 1.2.2 Lenguaje de bajo nivel 1.2.3 Lenguaje de alto nivel
	1.3 Lenguaje de Programación Python
	1.3.1 Instalación del lenguaje de programación Python y sus librerías (Windows y sistemas Unix)
	1.3.2 Uso de iteraciones (ciclo For – ciclo While)
	1.3.3 Manejo de archivos en Python (lectura/escritura)
	1.3.4 Manejo de estructuras de datos básicas en Python (Listas y Diccionarios)
	1.3.5 Implementación de algoritmos de ordenamiento y búsqueda en Python
	1.3.6 Uso de Graphviz
2. Jerarquía de Chomsky	2.1. Clasificación de gramáticas según restricciones 2.1.1 Lenguajes recursivamente enumerables 2.1.2 Lenguajes dependientes al contexto 2.1.3 Lenguajes independientes de contexto 2.1.4 Lenguajes regulares
	2.2. Ejemplos de gramáticas (Tipo 0 a 3)
3. Análisis léxico	3.1 Diseño de expresiones regulares y reconocimiento de tokens en Python (sin utilizar RegEx)
	3.2 Implementación y reconocimiento de tokens usando autómatas Finitos Deterministas (AFD)
	3.3 Autómatas Finitos No Deterministas (AFND)
	3.4 Técnicas de recuperación de errores léxicos (Modo pánico, eliminación de caracteres extraños)
4. Análisis sintáctico	4.1 Gramáticas libres de contexto
	4.2 Implementación de una gramática tipo 2 como analizador sintáctico
	4.3 Tabla de símbolos y tabla de errores.

	4.4 Ambigüedad y eliminación de ambigüedad
	4.5 Implementación de un autómata de pila
	4.6 Modo pánico como técnica de recuperación de errores sintácticos

BIBLIOGRAFÍA:

- Aho, A., Lam, M., Sethi, R. & Ullman, J. (2007). **Compilers: Principles, Techniques & Tools** (2nd ed.). Pearson.
- 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.
- Referencia del lenguaje Python. (2021). Python. <https://docs.python.org/es/3/reference/>