

Programa del curso. Laboratorio de Lenguajes Formales y de Programación			
Código:	0796	Créditos:	3
Escuela:	Ciencias y Sistemas	Área a la que pertenece:	Ciencias de la computación
Pre requisitos:	0770 - Introducción a la programación 1 0795 - Lógica de sistemas 0960 - Matemática de Cómputo 1	Post requisitos:	0777 - Organización de lenguajes y compiladores 1 0772 - Estructuras de datos
Categoría:	Obligatorio	Semestre	Segundo 2,021
Catedrático(a):	Ing. DAMARIS CAMPOS GONZALEZ	Auxiliar:	CÉSAR DAVID JUÁREZ GONZÁLEZ
Edificio:	Modalidad virtual	Sección:	A-
Salón del curso:	Salón 20 - Google Meet	Salón del laboratorio:	Salón 21 - 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	Días que se imparte el laboratorio:	Sábado
Horario del curso:	07:10 - 08:50 Horas	Horario del laboratorio:	12:10 - 13:50 Horas

Descripción del curso

Este laboratorio busca introducir al estudiante de la carrera al estudio, análisis y comprensión de los lenguajes de programación bajo una estructura genérica que contribuya a que el estudiante sea capaz de diseñar y comprender gramáticas, además, de los modelos matemáticos que las resuelven; abarcando conocimientos de lenguajes reales conocidos.

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 regulares y gramáticas.

Objetivos específicos

- Diseñar gramáticas que representen lenguajes específicos.
- Aplicar los conocimientos adquiridos en clase para implementar soluciones en el lenguaje de programación que se indique en laboratorio.

Metodología

- Se impartirán clases magistrales para la introducción de nuevos conceptos a los estudiantes.
- 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 y consideraciones

- La nota de promoción del laboratorio es de 61 puntos como mínimo.
- Es obligatorio aprobar el laboratorio para tener derecho a examen final en el curso.
- Se calificarán exámenes, prácticas y proyectos únicamente a estudiantes asignados en el curso.
- **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 Python**
- Copias parciales o totales en tareas, investigaciones, etc. Serán sancionadas con una nota de cero puntos.
- Las copias totales o parciales en los proyectos y práctica serán sancionadas con una nota de cero puntos. Los estudiantes involucrados serán reportados a la Escuela de Ciencias y Sistemas.

Evaluación

Aspecto	Valor
<ul style="list-style-type: none">• 4 Tareas - 1.5 puntos c/u	6 puntos
<ul style="list-style-type: none">• 2 Hojas de trabajo (2 puntos c/u)	4 puntos
<ul style="list-style-type: none">• Prácticas:<ul style="list-style-type: none">◦ Práctica 1: Publicación: 29/07/2021 Entrega: 19/08/2021	15 puntos
<ul style="list-style-type: none">• Proyectos<ul style="list-style-type: none">• Proyecto 1: Publicación: 20/08/2021 Entrega: 23/09/2021• Proyecto 2: Publicación: 24/09/2021 Entrega: 28/10/2021	30 puntos 35 puntos
<ul style="list-style-type: none">• Examen final (30/10/2021)	10 puntos
TOTAL	100 puntos



Tabla de contenido	
1. Lenguajes Formales	1.3 Lenguaje de Programación Python
	1.3.1 Instalación del lenguaje de programación Python. (Windows y sistemas Unix).
	1.3.2 Manejo de archivos en Python.
	1.3.3 Manejo de estructuras de datos básicas en Python (Listas y Diccionarios).
	1.3.4 Implementación de algoritmos de ordenamiento y búsqueda en Python
	1.3.5 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
	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
	4.2.1 Tabla de símbolos y tabla de errores.
	4.3 Modo pánico como técnica de recuperación de errores sintácticos

Bibliografía
1. Aho, Alfred V., Sethi y Ullman. <i>Compiladores: principios, técnicas y herramientas</i> (2nd ed.). Pearson.
2. Linz, P. (2017). <i>An Introduction to Formal Languages and Automata</i> (6th ed.). Jones & Bartlett Learning.
3. Hopcroft, John. y Ullman, Jeffrey (2008). <i>Introducción a la Teoría de Autómatas, Lenguajes y Computación.</i> (3rd ed.). Pearson.
4. Referencia del lenguaje Python. (2021). Python. https://docs.python.org/es/3/reference/