



FICHA TÉCNICA DEL CURSO: Lenguajes Formales y de Programación

No.	Descripción		Créditos
1	<b>Código</b> 796		<b>3</b>
2	<b>Escuela</b> Ciencias y Sistemas <a href="https://dtc-ecys.org">https://dtc-ecys.org</a>	<b>Área a la que pertenece</b> Ciencias de la computación	<b>Vigencia</b> Primer Semestre 2025
3	<b>2 períodos por semana</b>	<b>Horario:</b> Martes 07:10 a 08:50 am.	
4	<b>Pre-requisitos:</b> 770 Introducción a la Programación 1 795 Lógica de sistemas 960 Matemática de Cómputo 1	<b>Postrequisitos:</b> 777 Organización de lenguajes y compiladores 1 772 Estructuras de datos	
5	Secciones: A+, A-, B+, B-		
6	<p>I. Descripción General</p> <p>Este curso busca introducir al estudiante con los fundamentos teóricos matemáticos y conceptos que fundamentan los lenguajes de programación.</p> <p>Se busca, además, definir los modelos matemáticos asociados a la representación de los diferentes tipos de lenguajes para luego implementar estos conceptos en lenguajes de programación.</p> <p>Es de primordial importancia que pueda reconocer cualquier tipo de gramática, pero, sobre todo, pueda manejar y diseñar gramáticas para lenguajes regulares y libres del contexto, además, de los modelos matemáticos que las resuelven. Adquiriendo conceptos y los pueda relacionar a los aspectos técnicos y prácticos conociendo su aplicación en lenguajes reales conocidos. Y como estos conceptos son base introductoria a los compiladores.</p> <p>Al finalizar el curso el estudiante estará en la capacidad de comprender la funcionalidad de los compiladores.</p> <p>II. Objetivos</p> <p><b>Objetivo General</b> Que el estudiante conozca 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.</p> <p><b>Objetivos Específicos</b></p> <p>Al final del curso el estudiante deberá:</p> <ol style="list-style-type: none"> <li>1. Definir cualquier lenguaje formal</li> <li>2. Reconocer las características que identifican a cualquier tipo de gramática.</li> <li>3. Manejar la terminología de los lenguajes formales y gramáticas.</li> <li>4. Conocer el modelo matemático que resuelve cada tipo de gramática.</li> <li>5. Diseñar gramáticas que representen lenguajes específicos.</li> <li>6. Conocer e implementar máquinas de estado finito.</li> <li>7. Diseñar e implementar gramáticas regulares.</li> <li>8. Reconocer las fases de un compilador.</li> <li>9. Desarrollar un analizador léxico con base a los autómatas finitos.</li> <li>10. Desarrollar un analizador sintáctico por medio de las gramáticas libres del contexto.</li> </ol>		

### III. Contenido

<ol style="list-style-type: none"> <li>1. Unidad 1: Introducción (Capítulo 1, libro: Compiladores de Aho)             <ol style="list-style-type: none"> <li>1.1. Procesadores de lenguaje</li> <li>1.2. La estructura de un compilador                 <ol style="list-style-type: none"> <li>1.2.1. Análisis léxico</li> <li>1.2.2. Análisis sintáctico</li> <li>1.2.3. Análisis semántico</li> <li>1.2.4. Generación de código intermedio</li> <li>1.2.5. Optimización de código</li> <li>1.2.6. Generación de código</li> <li>1.2.7. Administración de la tabla de símbolos</li> <li>1.2.8. El agrupamiento de fases en pasadas</li> <li>1.2.9. Herramientas de construcción de compiladores</li> </ol> </li> <li>1.3. La evolución de los lenguajes de programación                 <ol style="list-style-type: none"> <li>1.3.1. El avance a los lenguajes de alto nivel</li> <li>1.3.2. Impactos en el compilador</li> </ol> </li> <li>1.4. La ciencia de construir un compilador                 <ol style="list-style-type: none"> <li>1.4.1. Modelado en el diseño e implementación de compiladores</li> <li>1.4.2. Aplicaciones de la tecnología de compiladores</li> </ol> </li> <li>1.5. Lenguajes formales                 <ol style="list-style-type: none"> <li>1.5.1. Definiciones (capítulo 1, libro de Linz)</li> <li>1.5.2. Jerarquía de Chomsky (capítulo 11, libro de Linz)</li> </ol> </li> </ol> </li> </ol>	<p>4 periodos</p>
<ol style="list-style-type: none"> <li>2. Unidad 2: Análisis léxico (Capítulo 3, libro: Compiladores de Aho)             <ol style="list-style-type: none"> <li>2.1. La función del analizador léxico                 <ol style="list-style-type: none"> <li>2.1.1. Comparación entre análisis léxico y análisis sintáctico</li> <li>2.1.2. Tokens, patrones y lexemas</li> <li>2.1.3. Atributos para los tokens</li> <li>2.1.4. Errores léxicos</li> </ol> </li> <li>2.2. Uso de búfer en la entrada                 <ol style="list-style-type: none"> <li>2.2.1. Pares de búferes</li> <li>2.2.2. Centinelas</li> </ol> </li> <li>2.3. Especificación de los tokens                 <ol style="list-style-type: none"> <li>2.3.1. Cadenas y lenguajes</li> <li>2.3.2. Operaciones en los lenguajes</li> <li>2.3.3. Definiciones regulares (Gramáticas regulares)</li> <li>2.3.4. Expresiones regulares</li> <li>2.3.5. Extensiones de las expresiones regulares</li> </ol> </li> <li>2.4. Reconocimiento de tokens                 <ol style="list-style-type: none"> <li>2.4.1. Diagrama de transición de estados</li> <li>2.4.2. Reconocimiento de las palabras reservadas y los identificadores</li> <li>2.4.3. Finalización del bosquejo</li> <li>2.4.4. Arquitectura de un analizador léxico basados en diagramas</li> </ol> </li> <li>2.5. Autómatas finitos                 <ol style="list-style-type: none"> <li>2.5.1. Autómatas finitos no deterministas</li> <li>2.5.2. Tablas de transición</li> </ol> </li> </ol> </li> </ol>	<p>12 periodos</p> <p><b>Parcial 1</b> <b>18-feb</b></p>

<ul style="list-style-type: none"> <li>2.5.3. Aceptación de las cadenas de entrada mediante los autómatas</li> <li>2.5.4. Autómatas finitos deterministas</li> <li>2.6. De las expresiones regulares a los autómatas <ul style="list-style-type: none"> <li>2.6.1. Conversión de un AFN a AFD</li> <li>2.6.2. Simulación de un AFN</li> <li>2.6.3. Eficiencia de la simulación de un AFN</li> <li>2.6.4. Construcción de una AFN a partir de una expresión regular</li> <li>2.6.5. Eficiencia de los algoritmos de procesamiento de cadenas</li> </ul> </li> <li>2.7. Diseño de un generador de analizadores léxicos <ul style="list-style-type: none"> <li>2.7.1. La estructura del analizador generado</li> <li>2.7.2. Coincidencia de patrones con base en los AFNs</li> <li>2.7.3. AFD para analizadores léxicos</li> <li>2.7.4. Implementación del operador de preanálisis</li> </ul> </li> <li>2.8. Optimización de los buscadores por concordancia de patrones basados en AFD <ul style="list-style-type: none"> <li>2.8.1. Estados significativos de una AFN</li> <li>2.8.2. Funciones calculadas a partir del árbol sintáctico</li> <li>2.8.3. Cálculo de anulable, primerapos y ultimapos</li> <li>2.8.4. Cálculo de siguientespos</li> <li>2.8.5. Conversión directa de una expresión regular a un AFD</li> <li>2.8.6. Minimización del número de estados de un AFD</li> <li>2.8.7. Minimización de estados en los analizadores léxicos</li> <li>2.8.8. Intercambio de tiempo por espacio en la simulación de un AFD</li> </ul> </li> </ul>	<p><b>Parcial 2</b> <b>18-marzo</b></p>
<p>3. Unidad 3: Análisis sintáctico (Capítulo 2, libro: Compiladores de Aho)</p> <ul style="list-style-type: none"> <li>3.1. Introducción</li> <li>3.2. Definición de sintaxis <ul style="list-style-type: none"> <li>3.2.1. Definición de gramáticas</li> <li>3.2.2. Derivaciones</li> <li>3.2.3. Árboles de análisis sintáctico</li> <li>3.2.4. Ambigüedad</li> <li>3.2.5. Asociatividad de los operadores</li> <li>3.2.6. Precedencia de los operadores</li> </ul> </li> <li>3.3. Autómata de Pila <ul style="list-style-type: none"> <li>3.3.1. Definición</li> <li>3.3.2. Notación gráfica</li> <li>3.3.3. Diseño de autómatas de pila</li> <li>3.3.4. Teorema 2.2: Generar AP desde una Gramática tipo 2</li> </ul> </li> <li>3.4. Análisis sintáctico</li> <li>3.5. Tabla de símbolos</li> </ul>	<p>8 periodos</p> <p><b>Parcial 3</b> <b>22-abril</b></p>

#### IV. Metodología:

El curso se desarrollará intercalando clases magistrales para la exposición de conceptos nuevos y clases participativas, en las que se espera que el estudiante realice las lecturas, tareas o ejercicios dejados para realizar fuera de clase, previo al inicio de un nuevo día de clase.

V. Evaluación:

La nota final estará compuesta de 100 puntos, distribuidos de la siguiente manera:

3 Evaluaciones de rendimiento (15 puntos c/u) .....	45 puntos
Tareas, trabajos en clase, comprobaciones, asistencia, cortos, etc .....	06 puntos
Laboratorio (proyectos, prácticas, etc.) .....	24 puntos
Evaluación Final .....	25 puntos
	-----
Nota Total .....	100 puntos

VI. Observaciones:

- Será necesario contar con un 80% de asistencia y aprobar el laboratorio del curso con una nota mínima de 61 puntos, para tener derecho a la evaluación final.
- Se realizarán exámenes cortos cada día de clase y se debe tener un 80% de exámenes realizados, para tener derecho a la evaluación final.
- 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.

7	Bibliografía	<ol style="list-style-type: none"> <li>1. Aho, A. V., Lam, M. S., Sethi, R., &amp; Ullman, J. D. (2007). <b>Compilers: Principles, Techniques &amp; Tools</b> (2nd ed.). Pearson.</li> <li>2. Linz, P. (2017). <b>An Introduction to Formal Languages and Automata</b> (6th ed.). Jones &amp; Bartlett Learning.</li> <li>3. Brookshear, J. Glenn. <b>Teoría de la Computación</b> - Lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana.</li> <li>4. Hopcroft, J. E., Motwani, R., &amp; Ullman, J. D. (2007). <b>Introduction Automata Theory, Languages and Computation</b> (3rd ed.). Pearson.</li> </ol>
8	No. De Secciones	4
9	Catedráticos titulares	Sección A+ Ing. Otto Rodríguez Sección B+ Ing. David Morales – <a href="mailto:demorales@ingenieria.usac.edu.gt">demorales@ingenieria.usac.edu.gt</a> Sección A- Inga. Damaris Campos – <a href="mailto:damaris.campos@ingenieria.usac.edu.gt">damaris.campos@ingenieria.usac.edu.gt</a> Sección B- Inga. Zulma Aguirre – <a href="mailto:zaguirre@ingenieria.usac.edu.gt">zaguirre@ingenieria.usac.edu.gt</a>
10	Director de Escuela	<b>Ing. Carlos Alonzo</b>