

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERIA**  
**ESCUELA DE CIENCIAS Y SISTEMAS**  
**PROGRAMA DE LABORATORIO DEL CURSO: Lenguajes Formales y de Programación.**

| No. | Descripción  |  |  |
|-----|--|--|--|
| .   | Código   | 796  | Créditos 3                               |
| 1   | <b>Escuela</b><br>Ciencias y Sistemas<br><a href="https://dti-ecys.org">https://dti-ecys.org</a>   | <b>Área a la que pertenece</b><br>Ciencias de la computación | <b>Vigencia</b><br>Segundo Semestre 2019 |
| 2   | <b>Horas clase</b><br>Martes 7;10-8:50   | <b>Horario Laboratorio</b><br>Martes 9:50 a 11:30            |  |
| 3   | <b>Pre-requisitos:</b><br>770 Introducción a la Programación 1<br>795 Lógica de sistemas<br>960 Matemática de Cómputo 1  |  |  |
| 4   | <b>Postrequisitos:</b><br>777 Organización de lenguajes y compiladores 1<br>772 Estructuras de datos   |  |  |
| 5   | <b>Sección: B+</b>   |  |  |
| 6   | <p><b>I. Descripción General</b></p> <p>Este curso busca introducir al estudiante con los fundamentos teóricos matemáticos y conceptos que fundamentan los lenguajes de programación. El estudiante debe adquirir la base teórica necesaria y requerida para que pueda llevar un curso avanzado de lenguajes y compiladores.</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 para lenguajes libres de contexto, además, de los modelos matemáticos que las resuelven. Se busca que el estudiante tenga mucha práctica en el diseño de gramáticas para representar lenguajes y que adquiera la habilidad de diseñarlas sin problema. Adquiriendo conceptos y los pueda relacionar a los aspectos técnicos y prácticos conociendo su aplicación en lenguajes reales conocidos. El estudiante debe aprender la teoría que esta atrás de los diferentes componentes de un compilador, las técnicas de programación usadas para poner esta teoría en practica, Al finalizar el curso el estudiante estará en la capacidad de poder recibir un curso avanzado de compiladores.</p> <p><b>II. Objetivos</b></p> <p><b>Objetivo General</b></p> <p>Que el estudiante tenga los conceptos teóricos y matemáticos necesarios, que fundamentan los lenguajes de programación y el diseño de lenguajes y compiladores.</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 y compiladores.</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 el funcionamiento de un analizador léxico y su implementación</li> <li>7. Conocer e implementar máquinas de estado finito</li> <li>8. Diseñar e implementar gramáticas libres de contexto</li> <li>9. Conocer los conceptos que fundamentan el análisis sintáctico</li> <li>10. Aplicar los modelos matemáticos que resuelven cualquier gramática</li> </ol> |  |  |

### III. Contenido

| Contenido  | Planificación |
|--|---------------|
| <b>Unidad 1. Lenguajes Formales</b> <ol style="list-style-type: none"><li>Definiciones<br/>Lenguajes de programación, Compiladores e interpretes<br/>Generaciones de lenguajes<br/>Partes del compilador: Análisis sintáctico<br/>Análisis léxico<br/>Definición de tokens, lexemas, palabras reservadas.</li><li>Lenguajes formales<ol style="list-style-type: none"><li>Definición</li><li>Símbolos terminales</li><li>Símbolos no terminales</li><li>Gramática</li><li>Estado inicial</li></ol></li></ol>   |               |
| <b>Unidad 2 : Jerarquía de Chomsky</b> <ol style="list-style-type: none"><li>Jerarquía de Chomsky<ol style="list-style-type: none"><li>Lenguajes Recursivamente enumerables<br/>Máquinas de Turing</li><li>Lenguajes Sensibles al Contexto<br/>Autómatas lineales limitados</li><li>Lenguajes Libres de contexto<br/>Autómatas Descendentes</li><li>Lenguajes regulares<br/>Autómatas finitos</li></ol></li></ol>  |               |
| <b>Unidad 3: Lenguajes regulares</b> <ol style="list-style-type: none"><li>Lenguajes regulares</li><li>Gramáticas Regulares (tipo 3)</li><li>Diseño de gramáticas regulares<ol style="list-style-type: none"><li>Ejemplos y ejercicios de gramáticas regulares</li><li>Aplicación de gramáticas regulares</li></ol></li><li>Expresiones regulares</li><li>Autómatas Finitos<ol style="list-style-type: none"><li>Grafos para representación de autómatas</li><li>Autómatas Finitos No Determinísticos NFA</li><li>Autómatas Finitos Determinísticos DFA</li></ol></li><li>Método para pasar de gramáticas a expresiones regulares y de expresiones regulares a gramáticas</li><li>Métodos para Calcular DFAs<ol style="list-style-type: none"><li>Construcción de Thomson y minimización de estados</li><li>Método del árbol</li></ol></li><li>Ejemplos y ejercicios</li></ol> |               |
| <b>Unidad 4: Lenguajes libres de contexto</b> <ol style="list-style-type: none"><li>Lenguajes Libres de contexto</li><li>Gramáticas Libres de contexto (Tipo 2)</li><li>Diseño de gramáticas libres de contexto<ol style="list-style-type: none"><li>Ejemplos y ejercicios de gramáticas libres de contexto</li><li>Aplicación de lenguajes libres de contexto</li></ol></li><li>Recursividad por la izquierda y recursividad por la derecha</li><li>Gramáticas ambiguas</li><li>Parser recursivos descendentes</li><li>Ejemplos y ejercicios</li></ol>  |               |

**IV. Metodología:**

El laboratorio será impartido 1 vez por semana, con implementación de material de apoyo que se le brindará a los estudiantes.

Se impartirán clases prácticas y teóricas para profundizar en los conocimientos de los estudiantes.

Resolución de problemas durante y fuera del laboratorio.

Elaboración de prácticas y proyectos.

Evaluación del conocimiento mediante tareas, cortos y examen final.

**V. Evaluación:**

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

|              |                         |
|--------------|-------------------------|
| Tareas       | 5 pts.                  |
| Cortos       | 5 pts.                  |
| Práctica     | 15 pts. (05-ago/23-ago) |
| Proyecto1    | 30 pts. (07-sep/04-oct) |
| Proyecto2    | 35 pts. (05-oct/08-nov) |
| Examen Final | 10 pts.                 |
| Total _____  | 100 pts.                |

**VI. Observaciones:**

- Será necesario contar con al menos un 81% de asistencia y aprobar el laboratorio del curso con una nota mínima de 61 puntos como mínimo.

|    |                     |   |
|----|---------------------|---|
| 7  | Bibliografía        | <ol style="list-style-type: none"> <li>1. L. Espino. Lenguajes Formales y Autómatas, 2015, Guatemala.</li> <li>2. Brookshear, J. Glenn. Teoría de la Computación - Lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana.</li> <li>3. Andrew W. Appel. Modern Compiler Implementation in Java. Second Edition. Cambridge University Press.</li> <li>4. Aho, Alfred V., Sethi y Ullman. Compiladores: principios, técnicas y herramientas. Addison-Wesley.</li> </ol> |
| 8  | No. De Secciones    | 4   |
| 9  | Catedrático titular | Sección B+ Ing. Luis Fernando Espino Barrios<br>usac.sistemas@gmail.com   |
| 10 | Auxiliar de catedra | Otto Guarchaj – ottoguitar@gmail.com  |
| 11 | Director de Escuela | Ing. Carlos Gustavo Alonzo  |