

FICHA TÉCNICA DEL CURSO: Lenguajes Formales y de Programación

| No. | Descripción | |
|-----|--|---|
| . | Código 796 | Créditos 3 |
| 1 | Escuela Ciencias y Sistemas https://dtt-ecys.org | Área a la que pertenece Ciencias de la computación Vigencia Primer Semestre 2020 |
| 2 | Horas por semana 2 | Horario Martes 7:00 a 8:40 |
| 3 | Pre-requisitos: 770 Introducción a la Programación 1 795 Lógica de sistemas 960 Matemática de Cómputo 1 | |
| 4 | Postrequisitos: 777 Organización de lenguajes y compiladores 1 772 Estructuras de datos | |
| 5 | Seccion: A+ | |
| 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. 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. Es de primordial importancia que pueda reconocer cualquier tipo de gramática, pero, sobre todo, pueda manejar y diseñar gramáticas para lenguajes regulares, 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. Al finalizar el curso el estudiante estará en la capacidad de poder recibir y comprender sin problema un curso avanzado de compiladores.</p> <p>II. Objetivos</p> <p>Objetivo General 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>Objetivos Específicos</p> <p>Al final del curso el estudiante deberá:</p> <ol style="list-style-type: none"> 1. Definir cualquier lenguaje formal 2. Reconocer las características que identifican a cualquier tipo de gramática. 3. Manejar la terminología de los lenguajes formales y gramáticas. 4. Conocer el modelo matemático que resuelve cada tipo de gramática. 5. Diseñar gramáticas que representen lenguajes específicos. 6. Conocer e implementar máquinas de estado finito. 7. Diseñar e implementar gramáticas regulares. | |

III. Contenido

| Contenido | Planificación |
|--|-------------------------------|
| <p>Unidad 1. Lenguajes Formales</p> <ol style="list-style-type: none"> 1. Definiciones Lenguaje natural, formal y de programación 2. Lenguajes formales <ol style="list-style-type: none"> 2.1 Definición 2.2 Alfabeto, símbolos, cadenas y reglas 2.3 Gramáticas en general 3. Lenguajes de programación <ol style="list-style-type: none"> 2.4 Definición 2.5 Breve historia y generaciones 2.6 Lenguaje de máquina, bajo nivel y alto nivel 2.7 Abstracciones de datos y control 2.8 Elementos de definición de lenguajes (sintaxis, semántica y pragmática) 4. Paradigmas de los lenguajes de programación <ol style="list-style-type: none"> 2.9 Imperativo <ol style="list-style-type: none"> 2.9.1 Procedimental 2.9.2 Estructurado 2.9.3 Orientado a Objetos 2.9.4 Programación concurrente o en paralelo 2.10 Declarativo <ol style="list-style-type: none"> 2.10.1 Funcional 2.10.2 Lógico 2.10.3 Orientado eventos 2.10.4 Lenguajes de Bases de datos | <p>● 4 períodos de clase</p> |
| <p>Unidad 2 : Jerarquía de Chomsky</p> <ol style="list-style-type: none"> 1. Jerarquía de Chomsky <ol style="list-style-type: none"> 1.1 Clasificación de gramáticas según restricciones 1.1 Lenguajes recursivamente enumerables Máquinas de Turing 1.2 Lenguajes dependientes al contexto Autómatas linealmente acotados 1.3 Lenguajes independientes de contexto Autómatas de pila 1.4 Lenguajes regulares Autómatas finitos Clasificación de los autómatas finitos | <p>● 2 períodos de clase</p> |
| <p>Unidad 3: Lenguajes regulares</p> <ol style="list-style-type: none"> 1. Lenguajes regulares 2. Autómatas Finitos Determinísticos DFA <ol style="list-style-type: none"> 2.1 Definición 2.2 Procesamiento 2.3 Notación y grafos para su representación 2.4 Ejemplos 3. Autómatas Finitos No Determinísticos NFA <ol style="list-style-type: none"> 3.1 Definición 3.2 Procesamiento 3.3 Notación y grafos para su representación 3.4 Ejemplos 4. Expresiones regulares <ol style="list-style-type: none"> 4.1 Definición 4.2 Operadores 4.3 Construcción 4.4 Precedencia 4.5 Ejemplos | <p>● 10 períodos de clase</p> |

| | |
|--|-----------------------|
| 5. Gramáticas Regulares (tipo 3) 5.1 Definición 5.2 Reglas 5.3 Derivación por la izquierda 5.4 Derivación por la derecha 5.5 Diseño de gramáticas regulares y ejemplos | |
| Unidad 4: Lenguajes independientes del contexto 1. Lenguajes Libres de contexto 2. Gramáticas Libres de contexto (Tipo 2) 2.1 Definición 2.2 Árboles de derivación 2.3 Diseño de gramáticas libres de contexto 2.4 Recursividad por la izquierda y recursividad por la derecha 3. Autómatas de pila 3.1 Definición 3.2 Notación gráfica 3.3 Diseño de autómatas de pila y procesamiento 3.4 Aceptación por estado final y por pila vacía 3.5 Ejemplos | ● 6 períodos de clase |

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. | 10 puntos |
| Laboratorio (proyectos, prácticas, etc.)..... | 20 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"> 1. Louden y Lambert. (2010). Lenguajes de programación. 3 ed. 2. Hopcroft, John. y Ullman, Jeffrey (2008). Introducción a la Teoría de Autómatas, Lenguajes y Computación. 3 ed. 3. Brookshear, J. Glenn. Teoría de la Computación - Lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana. 4. |
| 8 | No. De Secciones | 4 |
| 9 | Catedráticos titulares y auxiliares | Sección A- Inga. Damaris Campos de López lfyp.usac@gmail.com Sección A+ Ing. Otto Rodríguez Sección B- Inga. Zulma Aguirre aguirrezulma@gmail.com Sección B+ |
| 10 | Director de Escuela | Ing. Carlos Alonzo |