UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
Escuela de Ciencias y Sistemas

FICHA TÉCNICA DEL CURSO: Lenguajes Formales y de Programación

<table>
<tr><td>No.</td><td colspan="3">Descripción</td></tr>
<tr><td>.</td><td>Código 796</td><td colspan="2">Créditos 3</td></tr>
<tr><td>1</td><td>Escuela
Ciencias y Sistemas https://dtt-ecys.org</td><td>Área a la que pertenece
Ciencias de la computación</td><td>Vigencia
Primer Semestre 2022</td></tr>
<tr><td>2</td><td>Horas por semana
2</td><td colspan="2">Horario Martes 7:10 a 8:50</td></tr>
<tr><td>3</td><td colspan="3">Pre-requisitos:
770 introducción a la Programación 1
795 lógica de sistemas
960 Matemática de Cómputo 1</td></tr>
<tr><td>4</td><td colspan="3">Post requisitos:
777 Organización de lenguajes y compiladores 1
772 Estructuras de datos</td></tr>
<tr><td>5</td><td colspan="3">Secciones: A+, A-, B+, B-</td></tr>
</table>

6

I. Descripción General

Este curso busca introducir al estudiante con los fundamentos teóricos matemáticos y conceptos que fundamentan los lenguajes de programación. Busca introducir a los estudiantes a los conceptos de lenguajes y compiladores.
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.

II. Objetivos

Objetivo General

Que el estudiante conozca los conceptos teóricos y matemáticos necesarios que fundamentan los lenguajes formales y de programación así como los compiladores; mediante la clasificación de gramáticas, y el diseño de lenguajes mediante autómatas, expresiones y gramáticas.

Objetivos Específicos

Al final del curso el estudiante deberá:

- Definir cualquier lenguaje formal
- Reconocer las características que identifican a cualquier tipo de gramática.
- Manejar la terminología de los lenguajes formales y gramáticas.
- Conocer el modelo matemático que resuelve cada tipo de gramática.
- Diseñar gramáticas que representen lenguajes específicos.
- Conocer e implementar máquinas de estado finito.
- Diseñar e implementar gramáticas regulares.

III. Contenido

Contenido	Planificación
1. Unidad 1: Introducción 1.1. Procesadores de lenguaje 1.2. La estructura de un compilador 1.2.1. Análisis léxico 1.2.2. Análisis sintáctico 1.2.3. Análisis semántico 1.2.4. Generación de código intermedio 1.2.5. Optimización de código 1.2.6. Generación de código 1.2.7. Administración de la tabla de símbolos 1.2.8. El agrupamiento de fases en pasadas 1.2.9. Herramientas de construcción de compiladores 1.3. La evolución de los lenguajes de programación 1.3.1. El avance a los lenguajes de alto nivel 1.3.2. Impactos en el compilador 1.4. La ciencia de construir un compilador 1.4.1. Modelado en el diseño e implementación de compiladores 1.4.2. Aplicaciones de la tecnología de compiladores 1.5. Lenguajes formales 1.5.1. Definiciones 1.5.2. Jerarquía de Chomsky	
2. Unidad 2: Análisis léxico 2.1. La función del analizador léxico 2.1.1. Comparación entre análisis léxico y análisis sintáctico 2.1.2. Tokens, patrones y lexemas 2.1.3. Atributos para los tokens 2.1.4. Errores léxicos 2.2. Uso de búfer en la entrada 2.2.1. Pares de búferes 2.2.2. Centinelas 2.3. Especificación de los tokens 2.3.1. Cadenas y lenguajes 2.3.2. Operaciones en los lenguajes 2.3.3. Definiciones regulares (Gramáticas regulares) 2.3.4. Expresiones regulares 2.3.5. Extensiones de las expresiones regulares 2.4. Reconocimiento de tokens 2.4.1. Diagrama de transición de estados 2.4.2. Reconocimiento de las palabras reservadas y los identificadores 2.4.3. Finalización del bosquejo 2.4.4. Arquitectura de un analizador léxico basados en diagramas 2.5. Autómatas finitos 2.5.1. Autómatas finitos no deterministas 2.5.2. Tablas de transición 2.5.3. Aceptación de las cadenas de entrada mediante los autómatas 2.5.4. Autómatas finitos deterministas 2.6. De las expresiones regulares a los autómatas 2.6.1. Conversión de un AFN a AFD	***•Primera evaluación***

	2.6.2. Simulación de un AFN 2.6.3. Eficiencia de la simulación de un AFN 2.6.4. Construcción de una AFN a partir de una expresión regular 2.6.5. Eficiencia de los algoritmos de procesamiento de cadenas 2.7. Diseño de un generador de analizadores léxicos 2.7.1. La estructura del analizador generado 2.7.2. Coincidencia de patrones con base en los AFNs 2.7.3. AFD para analizadores léxicos 2.7.4. Implementación del operador de preanálisis 2.8. Optimización de los buscadores por concordancia de patrones basados en AFD 2.8.1. Estados significativos de una AFN 2.8.2. Funciones calculadas a partir del árbol sintáctico 2.8.3. Cálculo de anulable, primera posición y última posición 2.8.4. Cálculo de siguiente posición 2.8.5. Conversión directa de una expresión regular a un AFD 2.8.6. Minimización del número de estados de un AFD 2.8.7. Minimización de estados en los analizadores léxicos 2.8.8. Intercambio de tiempo por espacio en la simulación de un AFD	• ***Segunda evaluación***	
	3. Unidad 3: Introducción a la sintaxis 3.1. Introducción 3.2. Definición de sintaxis 3.2.1. Definición de gramáticas 3.2.2. Derivaciones 3.2.3. Árboles de análisis sintáctico 3.2.4. Ambigüedad 3.2.5. Asociatividad de los operadores 3.2.6. Precedencia de los operadores 3.3. Traducción orientada a la sintaxis 3.3.1. Notación prefija 3.3.2. Atributos sintetizados 3.3.3. Definiciones simples orientadas a la sintaxis 3.3.4. Recorrido de árboles 3.3.5. Esquemas de traducción		
	4. Unidad 4: Análisis sintáctico 4.1.1. Análisis sintáctico tipo arriba-abajo 4.1.2. Análisis sintáctico predictivo 4.1.3. Cuando usar las producciones épsilon 4.1.4. Diseño de un analizador sintáctico predictivo 4.1.5. Recursividad por la izquierda 4.2. Un traductor para las expresiones simples 4.2.1. Sintaxis abstracta y concreta 4.2.2. Adaptación del esquema de traducción 4.2.3. Procedimientos para los no terminales 4.2.4. Simplificación del traductor 4.2.5. El programa completo 4.3. Tabla de símbolos 4.3.1. Tabla de símbolos por alcance 4.3.2. El uso de las tablas de símbolos	• ***Tercera evaluación***	

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:

Descripción	Puntos
3 evaluaciones de rendimiento (15 puntos c/u).	45
Tareas, trabajos en clase, comprobaciones, asistencia, cortos, etc.	10
Laboratorio (incluye 2 proyectos de programación.)	20
Evaluación Final	25
Nota total	100

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 debe tener un 80% actividades realizadas e n clase.
- 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	1. Alfred V. Aho, Mónica S. Lam, Ravi Sethi, Jeffrey D. Ullman - **Compiladores: Principios, Técnicas y Herramientas** Pearson Addison Wesley 2006 2. Peter Linz **Una Introducción a los Lenguajes Formales y Autómatas** [6th ed. Jones & Bartlett 2017 3. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman **Introducción a la Teoría de Autómatas y lenguajes y Computación** Prentice Hall. 2006
8	No. De Secciones	4
9	Catedráticos	Sección A+ Ing. Otto Rodríguez Sección B+ Ing. David Morales Sección A- Inga. Damaris Campos Sección B- Inga. Zulma Aguirre
10	Director de Escuela	**Ing. Carlos Alonzo**