UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
ESCUELA DE CIENCIAS Y SISTEMAS

NOMBRE DEL CURSO: Lenguajes Formales y de Programación

CÓDIGO:	**796**	**CRÉDITOS:**	**3**
ESCUELA:	**Ciencias y Sistemas**	**ÁREA A LA QUE PERTENECE:**	**Ciencias de la computación**
PRE REQUISITOS:	**770 – Introducción a la computación y programación 1** **795 – Lógica de sistemas** **960 – Matemática para la computación 1**	**POST REQUISITOS:**	**777 – Organización de lenguajes y compiladores 1** **964 – Organización computacional**
CATEGORÍA:	**Obligatorio**	**SEMESTRE:**	**2do semestre 2014**
CATEDRÁTICO (A):	**Inga. Damaris Campos de López**	**AUXILIAR:**	**Erick Carlos Roberto Navarro Delgado**
HORARIO DEL CURSO:	**Martes 07:00 AM – 08:40 AM**	**HORARIO DEL LABORATORIO:**	**Viernes de 13:10 a 14:50**

Descripción del curso:

Este laboratorio tiene como propósito poner en práctica los conocimientos teóricos adquiridos en el curso de Lenguajes Formales.

Durante el laboratorio, el estudiante aprenderá a implementar correctamente los elementos básicos de un compilador que están involucrados en las fases de análisis léxico, sintáctico y semántico.

El estudiante deberá diseñar gramáticas que reconozcan los lenguajes que se proponen en las prácticas y proyectos del laboratorio, y utilizar diversas técnicas y métodos de programación, para implementarlas.

Objetivo General:

Que el estudiante obtenga los conocimientos teóricos y prácticos que fundamentan el diseño de lenguajes de programación y compiladores.

Objetivos Específicos:

- Que el estudiante sea capaz de utilizar los conceptos teóricos para implementar un lenguaje formal.
- Que el estudiante sea capaz de desarrollar analizadores léxicos en base a autómatas finitos deterministas.
- Que el estudiante sea capaz obtener autómatas finitos deterministas a partir de expresiones regulares mediante el método de Thompson y la construcción de subconjuntos.
- Que el estudiante pueda obtener autómatas finitos deterministas a partir de expresiones regulares mediante el método del árbol.

- Que el estudiante sea capaz de desarrollar analizadores sintácticos en base a gramáticas independientes del contexto.
- Que el estudiante pueda implementar un analizador sintáctico predictivo recursivo, en base a una gramática independiente del contexto que cumpla con las restricciones correspondientes.

Metodología:

Se impartirán clases presenciales en las cuales se complementará el contenido teórico del curso, poniendo en práctica los conocimientos adquiridos mediante ejercicios y otras actividades prácticas.

Adicionalmente se realizarán proyectos, prácticas, tareas, exámenes cortos y hojas de trabajo con el objetivo de mejorar las habilidades de los estudiantes en el diseño y construcción de compiladores y evaluar los conocimientos adquiridos.

Dentro del desarrollo del laboratorio el alumno deberá tomar en cuenta lo siguiente:

- Copias parciales o totales en las tareas, investigaciones, etc. serán sancionadas.
- Copias en los proyectos y prácticas serán sancionadas y reportadas a la escuela de sistemas.
- Las tareas, investigaciones, prácticas, proyectos, etc., deben ser entregadas en la fecha indicada y con el formato establecido.

Evaluación:

La nota final del laboratorio, se distribuye en actividades de evaluación de la siguiente manera:

Tareas, exámenes cortos y hojas de trabajo……	5 puntos
Práctica 1 *(5 ago-23 ago)*……	10 puntos
Primer proyecto *(26 ago-20 sep)*……...	30 puntos
Práctica 2 *(23 sep-7 oct)*……..…….……	10 puntos
Segundo proyecto *(7 oct-4 nov)*…….	35 puntos
Examen Final…….	10 puntos
Total……	**100 puntos**

Observaciones:

- El laboratorio se debe de aprobar con una nota mínima de 61 puntos.
- Solo se calificaran exámenes, proyectos y demás actividades, a estudiantes asignados.
- **Las prácticas y proyectos deben desarrollarse utilizando Visual Basic versión 2013**.

Contenido:

1. Conceptos generales
    a. Definición de lenguaje formal
    b. Definición de compilador
    c. Fases del compilador
    d. Definición de intérprete
    e. Diferencias entre compiladores e intérpretes
2. Análisis Léxico
    a. Token, lexema, patrón y ejemplos
    b. Implementación de un analizador léxico
    c. Tabla de símbolos
3. Jerarquía de Chomsky
    a. Niveles
    b. Restricciones
    c. Ejemplos
4. Lenguajes regulares y gramáticas regulares
    a. Definición
    b. Diseño y construcción
    c. Arboles de derivación
    d. Ejemplos y aplicaciones
5. Expresiones Regulares
    a. Definición y Propiedades
    b. Diseño y construcción
    c. Relación Gramáticas Regulares – Expresiones Regulares
6. Autómatas Finitos
    a. Definición
    b. Tabla de transición
    c. Autómatas Finitos No Deterministas
    d. Autómatas Finitos Deterministas
    e. Implementación de Autómatas Finitos Deterministas
7. Método del Árbol
    a. Construcción del árbol
    b. Calculo de primeros, últimos y siguientes.
    c. Construcción del autómata finito determinista
    d. Implementación del método del árbol
    e. Resolución de ejercicios aplicando el método
8. Método de Thompson y subconjuntos
    a. Nomenclatura
    b. Resolución de ejercicios aplicando el método
9. Análisis Sintáctico
    a. Funcionamiento de un analizador sintáctico
    b. Implementación de analizadores sintácticos
    c. Análisis sintáctico ascendente
    d. Análisis sintáctico descendente

10. Gramáticas independientes del contexto
    a. Árboles de derivación
    b. Ambigüedad y recursividad
    c. Factorización por la izquierda
    d. Eliminación de la recursividad por la izquierda
    e. Implementación de autómatas de pila
    f. Analizador sintáctico descendente
11. Analizador sintáctico descendente LL(1)
    a. Primeros y siguientes
    b. Construcción de tabla de análisis sintáctico

Bibliografía:

- Aho, Alfred V., Sethi y Ullman. Compiladores: principios, técnicas y herramientas. Addison-Wesley.
- Brookshear, J. Glenn. Teoría de la Computación - Lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana.
- Andrew W. Appel. Modern Compiler Implementation in Java. Second Edition. Cambridge University Press.
- Hopcroft, John y Ullman, Jeffrey. Introducción a la Teoría de Autómatas, Lenguajes y Computación.
- Kenneth Louden. Construcción de compiladores