UNIVERSIDAD DE SAN CARLOS DE GUATEMALA 

FACULTAD DE INGENIERIA

ESCUELA DE SISTEMAS

|  |
| --- |
| **NOMBRE DEL CURSO: Laboratorio de Lenguajes Formales y de Programación** |
| **CODIGO:**  | **796** | **CREDITOS:** | **3** |
| **ESCUELA:**  | **Ciencias y Sistemas** | **AREA A LA QUE PERTENECE:**  | **Ciencias de la computación** |
| **PRE REQUISITOS:**  | **770 – Introducción a la Programación 1****795 – Lógica de sistemas****960 – Matemática de cómputo 1** | **POST REQUISITO:**  | **777 Organización de Lenguajes y Compiladores 1** **772 Estructuras de Datos** |
| **CATEGORIA:**  | **Obligatorio** |  **SEMESTRE:** | **2do. 2017** |
| **CATEDRÁTICO (A):** | **Inga. Damaris Campos de López** | **AUXILIAR:**  | **Nestor Alexander Tzunún Mérida** |
| **EDIFICIO:**  | **T-3** | **SECCIÓN:** | **A-** |
| **SALON DEL CURSO:**  | **215** | **SALON DEL LABORATORIO:**  | **T-3 210** |
| **HORAS POR SEMANA DEL CURSO:**  | **2** | **HORAS POR SEMANA DEL LABORATORIO:** | **2** |
| **DÍAS QUE SE IMPARTE EL CURSO:** | **Martes** | **DIAS QUE SE IMPARTE EL LABORATORIO:** | **Lunes** |
| **HORARIO DEL CURSO:** | **07:00 – 08:40**  | **HORARIO DEL LABORATORIO:** | **07:10 – 08:50** |

**DESCRIPCIÓN DEL CURSO:**

El laboratorio tiene como propósito introducir al estudiante de ciencias de la computación al estudio, análisis, comprensión e implementación de lenguajes de programación bajo una estructura genérica que contribuya al desarrollo de un compilador básico y funcional; abarcando las fases de análisis léxico, análisis sintáctico e introducción al análisis semántico.

Comprende también mostrar métodos que faciliten la creación de analizadores léxicos por medio de algoritmos programados.

**OBJETIVOS:**

***Objetivo General***

***•*** Introducir al estudiante al conocimiento y desarrollo de las funciones básicas de los compiladores, y los ponga en práctica en la construcción de las primeras fases de este.

***Objetivos Específicos***

***•*** Aplicar los conocimientos adquiridos en clase para implementar analizadores léxicos.

***•*** Implementar un analizador sintáctico utilizando las técnicas vistas en clase.

**METODOLOGIA:**

• Se imparten clases presenciales con material de apoyo que es proporcionado al estudiante al finalizar la clase.

• En el transcurso del semestre se realizarán tareas y exámenes cortos para evaluar los conocimientos adquiridos.

• Se realizan prácticas y proyectos en donde se ponga en práctica y se puedan evaluar los conceptos adquiridos en el curso, tomando en cuenta que pueden incluirse temas de cursos pre requisito.

• Se impartirán clases prácticas donde se resuelvan problemas relacionados con el tema a desarrollar.

### REQUISITOS:

### El laboratorio se debe aprobar con nota mínima de 61 puntos.

### Es obligatorio aprobar el laboratorio para tener derecho a examen final.

### Solo se calificarán exámenes y proyectos de estudiantes asignados en el curso. NO se agreganestudiantes a actas

### Será necesario contar con un 80% de asistencia.

### 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.

### Las prácticas y proyectos deben desarrollarse utilizando lenguaje C#

### Copias parciales o totales en las tareas, investigaciones. serán sancionadas con una nota de cero.

### Copias en los proyectos y prácticas serán sancionadas con una nota de cero y reportadas a la Escuela de Sistemas.

### Las tareas, investigaciones, prácticas, proyectos,deben ser entregadas en la fecha indicada y con el formato establecido.

**EVALUACION:**

**Actividad Valor**

Tareas, cortos y hojas de trabajo 10 pts.

Práctica 1 15 pts.

 Publicación: 11/02/2018

 Entrega: 26/02/2018

Proyecto 1 30 pts

Publicación: 4/03/2018

 Entrega: 2/04/2018

Proyecto 2 35 pts

Publicación: 2/04/2018

 Entrega: 4/05/2018

Examen final 10pts

 ***Total 100 pts.***

\*Para tener derecho a examen final se requiere una asistencia mayor o igual a 80%.

**CONTENIDO**

**1. Lenguajes**

1.1 Lenguajes Naturales

 1.2 Características de lenguajes naturales

1.3 Lenguajes Formales

1.4 Características de lenguajes formales

1.5 Lenguajes de programación

**2. Introducción a compiladores**

2.1 ¿Qué es un compilador?

2.2 Ejemplos de compiladores

2.3 Diferencia entre compilador e intérprete

2.4 Partes del compilador

**3. Análisis léxico**

3.1 Análisis lexicográfico

3.2 Token, Patrón y Lexema

3.3 Errores léxicos

**4. Jerarquía de Chomsky**

4.1 Tipos de gramáticas según Chomsky

4.2 Restricciones

4.3 Ejemplos

**5. Lenguajes regulares y gramáticas regulares**

5.1 Lenguajes regulares y gramáticas

5.2 Expresiones regulares

5.3 Ejemplos y aplicaciones

**6. Autómatas finitos**

6.1. Definición

6.2. Tabla de transiciones

6.3. Autómatas Finitos Deterministas (AFD)

6.4. Autómatas Finitos No Deterministas (AFND)

6.5. Implementación de AFD’s

**7. Método de Thompson**

7.1. Nomenclatura

7.2. Cerraduras y subconjuntos

7.3. Transiciones

**8. Método del árbol**

8.1. Construcción de árboles

8.2. Cálculo de primera, últimas y siguiente posición

8.3. Subconjuntos y transiciones

8.4. Implementación del método del árbol

**9.Implementación de un analizador léxico**

 9.1 Tabla de símbolos

9.2 Programación de un autómata finito

 9.3 Manejo de errores

**10. Lenguajes libres de contexto**

10.1 Lenguajes Independientes del contexto

10.2 Gramáticas Tipo 2

10.3 Diseño de gramáticas independientes del contexto

10.4Ejemplos y ejercicios de gramáticas libres de contexto

10.5 Recursividad

 10.5.1 Por la izquierda

 10.5.2 Por la derecha

10.6 Gramáticas ambiguas

10.7Implementación de autómatas de pila

10.8Parser recursivo descendente

10.9 Ejemplos y ejercicios

**BIBLIOGRAFÍA**:

* Aho, Alfred V., Sethi y Ullman. Compiladores: principios, técnicas yherramientas. Addison-Wesley.
* Brookshear, J. Glenn. Teoría de la Computación - Lenguajes formales, autómatas y complejidad. Addison-Wesley Iberoamericana.
* John E Hopcroft. introducción a la Teoría de Autómatas, Lenguajes y computación.