

**NOMBRE DEL CURSO: Organización de Lenguajes y Compiladores 1**

<b>CODIGO:</b>	0777	<b>CREDITOS:</b>	4
<b>ESCUELA:</b>	Ciencias y Sistemas	<b>AREA A LA QUE PERTENECE:</b>	Ciencias de la Computación
<b>PRE REQUISITO:</b>	0771 – Introducción a la Programación y Computación 2 0796 – Lenguajes Formales y de Programación 0962 – Matemática para Computación 2	<b>POST REQUISITO:</b>	Organización de Lenguajes y Compiladores 2 [781]
<b>CATEGORIA:</b>	Obligatorio	<b>SEMESTRE:</b>	2do. 2014
<b>CATEDRÁTICO (A):</b>	Ing. Mario José Bautista Fuentes	<b>AUXILIAR:</b>	Eduardo López
<b>EDIFICIO:</b>	T-3	<b>SECCIÓN:</b>	C
<b>SALON DEL CURSO:</b>	113	<b>SALON DEL LABORATORIO:</b>	T3 - 113
<b>HORAS POR SEMANA DEL CURSO:</b>	4	<b>HORAS POR SEMANA DEL LABORATORIO:</b>	2
<b>DÍAS QUE SE IMPARTE EL CURSO:</b>	Sábado	<b>DIAS QUE SE IMPARTE EL LABORATORIO:</b>	Sábado
<b>HORARIO DEL CURSO:</b>	10:50AM – 14:10AM.	<b>HORARIO DEL LABORATORIO:</b>	10:50 – 12:30

**DESCRIPCIÓN DEL CURSO:**

Organización de Lenguajes y Compiladores 1 implementa los principios básicos de un compilador y/o intérprete, tomando como referencia diferentes herramientas de programación, describiendo las fases de cada proceso.

En el laboratorio se trata en detalle las primeras fases del proceso: análisis lexicográfico, análisis sintáctico. Así como también el manejo de errores y de la tabla de símbolos para así poner en práctica los conceptos aprendidos en clase como en el laboratorio.

**OBJETIVO GENERAL:**

Que el estudiante adquiera experiencia en el uso de las diferentes tipos de herramientas de programación para el proceso de compilación.

**Objetivos Específicos:**

1. Que el estudiante aprenda en detalle las primeras fases del proceso de compilación, principalmente el análisis lexicográfico, el análisis sintáctico y análisis semántico.
2. Capacitar y ejercitar al estudiante en los principios del análisis, diseño e implementación de compiladores, para lo cual se realizarán varias tareas y proyectos.
3. Que el estudiante conozca las diferentes herramientas para realizar análisis léxico y sintáctico.

**METODOLOGIA:**

- Clase magistral una vez por semana donde será explicada la teoría.
- Resolución de tareas y evaluaciones.
- Proyectos de programación.
- Autoestudio.

**REQUISITOS:**

- Todas las prácticas y proyectos deben de tener una documentación completa, clara y específica.
- Para poder aprobar el curso, es obligatorio aprobar el laboratorio, la nota mínima de promoción es con 61/100.

**EVALUACION:**

<b>Aspecto</b>	<b>Valor (pts.)</b>
Primer Proyecto	30
Segundo Proyecto	40
Practicas	10
Tareas y hojas de trabajo	10
Evaluación Final	10
	-----
Total de laboratorio	100

**CONTENIDO**

Las unidades que el laboratorio comprende, se dividen en 9 unidades:

**1. Procesador de Lenguaje.**

- 1.1 Diferencia entre intérprete y compilador.
- 1.2 Tipos de compiladores.
- 1.3 Fases de análisis.
- 1.4 Fase de síntesis.
- 1.4 Tabla de símbolos.
- 1.5 Detección de errores.

## **2. Análisis Léxico.**

- 2.1 *Análisis lexicográfico*
- 2.2 *Función del analizador.*
- 2.3 *Componentes léxicos, patrones, lexemas.*
- 2.4 *Autómata.*
- 2.4 *Expresiones regulares.*
- 2.4 *Tratamiento de errores léxicos.*

## **3. Analizador Léxico (JLex/JFlex).**

- 3.1 *Código de usuario.*
- 3.2 *Directivas del analizador.*
- 3.3 *Reglas de expresiones regulares.*
- 3.4 *Acciones léxicas.*

## **4. Análisis Sintáctico.**

- 4.1 *Análisis de sintaxis.*
- 4.2 *Gramáticas independientes de contexto.*
- 4.3 *Derivación.*
- 4.4 *Ambigüedad.*
- 4.5 *Árbol del análisis sintáctico.*

## **5. CUP – Constructor of Useful Parsers.**

- 5.1 *Historia.*
- 5.2 *Funcionamiento.*
- 5.3 *Descripción de clases generadas por Cup.*
- 5.4 *Código de usuario.*
- 5.5 *Declaración de símbolos gramaticales.*
- 5.6 *Declaración de precedencia.*
- 5.7 *Definición de gramáticas.*
- 5.8 *Métodos de recuperación de errores.*

## **6. Gramáticas Ascendentes.**

- 6.1 *Gramáticas.*
- 6.2 *Gramáticas ascendentes.*
- 6.3 *Funcionamiento.*
- 6.4 *Implementar pila por desplazamiento y reducción.*
- 6.5 *Uso de precedencia de operadores.*
- 6.6 *Ejemplos de gramáticas.*

## **7. Gramáticas Descendentes.**

- 7.1 Funcionamiento*
- 7.2 Reescritura de gramáticas*
- 7.3 Supresión de la ambigüedad.*
- 7.4 Factorización por la izquierda.*
- 7.5 Eliminación de la recursiva por la izquierda.*

## **8. Tabla de Símbolos.**

- 8.1 Tabla de símbolos.*
- 8.2 Funcionamiento.*
- 8.3 Compilador de una pasada.*
- 8.4 Compilador de varias pasadas.*
- 8.5 Contenido de la tabla de símbolo.*
- 8.6 Operaciones de la tabla de símbolos.*

## **9. Traducción dirigida por la sintaxis.**

- 9.1 Definición dirigida por la sintaxis.*
- 9.2 Construcción del árbol de traducción dirigida por la sintaxis.*
- 9.3 Definición de atributos.*
- 9.4 Tipos de atributos.*
- 9.5 Atributos Sintetizados*
- 9.6 Ejemplo gramática con atributos sintetizados.*
- 9.7 Atributos heredados.*
- 9.8 Ejemplo gramática atributos heredados.*
- 9.9 Esquema de traducción.*
- 9.10 Evaluación de atributos.*

## **BIBLIOGRAFÍA:**

- Compiladores. Principios, Técnicas y Herramientas Aho, Sethi y Ullman. Addison-Wesley Iberoamericana, 1990