Nombre de la Asignatura: **PROCESADORES DE LENGUAJES I** Código: **404**
Créditos Teóricos: **3** Créditos Prácticos: **2.5** Créditos Totales:**5.5**
Nombre del Departamento: **Estadística, Investigación Operativa y Computación.**
Profesor que imparte la asignatura: **Coromoto León Hernández** e-mail: **coro@csi.ull.es**
Teléfono: 922 31 81 80
Horario de Tutorías: **Lunes y Miércoles 9:00-12:00**
Horario de Clases: **Lunes, Marte y Miércoles: 15:55-16:50**
Número de Horas Prácticas: **1 hora semanal (Sala A)** Número de Grupos: **4**
Horario de Prácticas: **Lunes: 15:00-15:55; Martes: 9:30-10:30, 11:00-12:00, 12:00-13:00**
Método de Evaluación:

*La nota de la asignatura está repartida de la siguiente manera: **80% Teoría + 20% Prácticas***
*La calificación obtenida en el examen teórico constituye el 80% de la nota final de la asignatura. Semanalmente se evalúan los ejercicios prácticos en el laboratorio, constituyendo la totalidad de los mismos el 20% de la nota final de la asignatura. Si no se supera la realización del **75%** de los Ejercicios Prácticos, se cuenta con un examen práctico final.*

Objetivos:

Entender el proceso de compilación. Comprender las técnicas involucradas en el análisis del lenguaje fuente. Conocer cual es la misión de un analizador léxico y saber implementarlo usando la herramienta LEX. Conocer el concepto de análisis sintáctico y saber implementar uno utilizando la herramienta YACC. Comprender el significado de la semántica de un lenguaje y su implementación mediante el análisis de ámbito y el análisis de tipo.

Fecha de los exámenes: **1/14 de Febrero, 10 de Junio, 5 de Septiembre**

Programa de Teoría:

1.- **Conceptos de lenguajes de programación.**
Clasificación de los lenguajes de programación. Evolución de los lenguajes de programación y su incidencia en los traductores. Conceptos de Lenguajes, sintaxis, semántica. Conceptos de traductores, compiladores e intérpretes. Esquema de un compilador.

2.- **Análisis léxico.**
Función del analizador léxico. Lexemas, expresiones regulares y tokens. Técnicas para la construcción del analizador léxico. Generación de analizadores léxicos. El generador de analizadores léxicos LEX. Expresiones regulares en LEX.

3.- **Análisis sintáctico.**
Clasificación de los analizadores sintácticos. Analizadores ascendentes. Analizadores LR. Generación de analizadores sintácticos. El generador de analizadores sintácticos YACC. Evaluación de atributos en YACC. Utilización conjunta de LEX y YACC.

4.- **Análisis semántico.**
Definiciones dirigidas por la sintaxis. Evaluación ascendente de definiciones con atributos sintetizados. Definiciones con atributos por la izquierda. Traducción descendente. Evaluación ascendente de atributos heredados. Sistemas de tipos: expresiones de tipos; equivalencia estructural, nominal y funcional; sobrecarga de operadores. Tablas de símbolos: estructura de la tabla de símbolos; organización de la tabla de símbolos; lenguajes con estructura de bloque.

5.- **Manipulación de Memoria.**
Organización de la memoria en tiempo de ejecución. La memoria estática. La pila. El heap.

6.- **Generación de Código Intermedio**.
Lenguajes intermedios. Declaraciones. Proposiciones de asignación. Expresiones booleanas. Proposiciones case. Relleno de retroceso. Llamadas a procedimientos.

Programa de Prácticas:

1.- **Título**: ***Modo de uso de la herramienta: lex.***
Objetivo: Aprendizaje de esta herramienta de construcción de analizadores léxicos para PCs.

2.- **Título**: ***Modo de uso de las herramienta: AT&T-lex y flex.***
Objetivo: Aprendizaje de estas herramienta de construcción de analizadores léxicos en el entorno Unix.

3.- **Título** : ***Reconocimiento de componentes léxicos. Aplicación.***
Objetivo : Construir un analizador léxico de una línea de comandos usando las herramientas MKS-lex, AT&T-lex y flex.

4.- **Título** : ***Reconocimiento de componentes léxicos. Aplicación.***
Objetivo : Construir un analizador léxico para un lenguaje sencillo usando un generador de analizadores léxicos.

5.- **Título** : ***Modo de uso de las herramientas: yacc, AT&T-yacc y bison.***
Objetivo : Aprendizaje de estas herramienta de construcción de analizadores sintácticos para PCs y entornos Unix.

6.- **Título** : ***Reconocimiento de lenguajes. Aplicación.***
Objetivo : Generar un analizador sintáctico para un lenguaje sencillo utilizando YACC.

7.- **Título** : ***Reconocimiento de lenguajes. Aplicación.***
Objetivo : Generar un analizador sintáctico para una lenguaje sencillo utilizando YACC y ampliarlo con acciones semánticas que permitan reformatear el fichero de entrada.

8.- **Título** : ***Recuperación de errores.***
Objetivo : Modificar el analizador sintáctico escrito para que sea capaz de recuperarse de los errores que se produzcan.

9.- **Título** : ***Restricciones semánticas.***
Objetivo : Añadir los atributos y acciones semánticas al analizador sintáctico.

10.- **Título** : ***Asignación de Memoria.***
Objetivo : Seleccionar e implementar las estructuras necesarias que permitan realizar la asignación de memoria.

Bibliografía:

[1] Aho, A.; Sethi, R.; Ullman, J., "COMPILERS. Principles, Techniques, and Tools", Adisson-Wesley Publishing company, 1986.

[2] Gardner, J.; Linseman, A.; Nicol, S.; Retterath C. "MKS LEX & YACC", Mortice Kern Systems Inc, 1992.

[3] Levine, J.; Mason, T.; Brown, D., "LEX & YACC", O'Reilly & Associates, Inc. 1990.

[4] León Hernández, C. “Prácticas de Compiladores en C y Pascal”. Gobierno de Canarias, 1997.

[5] Schreiner, A.; Friedman, Jr. H., "INTRODUCTION TO COMPILER CONSTRUCTION WITH UNIX", Prentice-Hall, 1985.