

**12717 - AMPLIACIÓN DE
PROCESADORES DE LENGUAJES**

ASIGNATURA: 12717 - AMPLIACIÓN DE PROCESADORES DE LENGUAJES

Vinculado a : (Titulación - Asignatura - Especialidad)

1801-Ingeniería en Informática - 12717-AMPLIACIÓN DE PROCESADORES DE LENGUAJES - 00

CENTRO: Escuela de Ingeniería Informática

TITULACIÓN: Ingeniero en Informática

DEPARTAMENTO: INFORMÁTICA Y SISTEMAS

ÁREA: Lenguajes Y Sistemas Informáticos

PLAN: 10 - Año 199**ESPECIALIDAD:**

CURSO: Cr. comunes cic**IMPARTIDA:** Segundo semestre **TIPO:** Optativa

CRÉDITOS: 9

TEÓRICOS: 4,5

PRÁCTICOS: 4,5

Descriptores B.O.E.

Lenguajes. Diseño. Generación de Códigos. Herramientas y Técnicas de Implementación. Estudio de Casos.

Temario

- I. INTRODUCCIÓN (~1h)
- II. ANÁLISIS SINTÁCTICO (~9h)
 - 1. Generadores descendentes
 - 2. Métodos de avance-reducción
 - 3. Métodos generales
- III. ANÁLISIS SEMÁNTICO (~6h)
 - 1. Árbol abstracto
 - 2. Gramáticas atribuidas
 - 3. Evaluación de atributos
 - 4. L- y S-atribución
- IV. GESTION DE MEMORIA (~3h)
 - 1. Garbage collection
- V. PARADIGMAS (~16h)
 - 1. Imperativo
 - 2. Orientado a objeto
 - 3. Paralelo y distribuido
 - 4. Funcional
 - 5. Lógico
- VI. OPTIMIZACIÓN (~10h)
 - 1. Local
 - 2. Global
 - 3. Asignación de registros
 - 4. Otros

Requisitos Previos

Conocimientos de lenguajes de programación, compiladores e intérpretes. Experiencia en programación con lenguajes de alto y de bajo nivel.

Objetivos

El objetivo principal es el de suplementar la formación que el alumno interesado ha adquirido en el conocimiento y diseño de procesadores y lenguajes, principalmente en las asignaturas de Compiladores y de Procesadores de Lenguajes. Se pretende que el alumno consolide y amplíe dichos conocimientos, mediante la exposición teórica de técnicas y elementos adicionales, y mediante una actividad práctica que incorpore diseño e implementación. El énfasis es principalmente práctico, junto con la presentación de los elementos teóricos fundamentales, mediante el uso de herramientas actuales, que posibiliten una continuidad en la autoformación en este campo del futuro titulado. Consideraremos asimismo importante fomentar la propia iniciativa, el espíritu crítico y la capacidad de exposición y debate de las ideas.

Metodología

Si bien se desea dotar al alumno de unos fundamentos teóricos sólidos, el enfoque de la asignatura es esencialmente práctico y realista. Dentro de este enfoque, se pone énfasis en la planificación, el seguimiento de la realización, la documentación, y su defensa. Pretendemos motivar al alumno en este aspecto central de la informática ofreciéndole flexibilidad (supervisada) en la especificación de los objetivos y el diseño, y con el fomento y valoración de la iniciativa, novedad y originalidad.

Se pondrá énfasis especial en la utilización de modernos entornos de generación y lenguajes de programación, como es el caso de JavaCC, Cup o ANTLR para Java, y la utilización de bytecodes y de implementación de aspectos esenciales de paradigmas como la orientación a objeto; así, no sólo se ofrecen al alumno diversas opciones sino incluso se fomenta su iniciativa, y la posibilidad de desarrollar los trabajos en grupo o individualmente.

Criterios de Evaluación

Para la superación del curso existirá una única prueba teórica recuperable (y sustituible por evaluación continua dependiendo del volumen de alumnado) y una evaluación del trabajo realizado, ambos necesarios y con igual ponderación final. En los trabajos se valorará el esfuerzo individual, la iniciativa, la aplicación de las técnicas y herramientas apropiadas, la orientación práctica y la regularidad en su desarrollo temporal.

Descripción de las Prácticas

Se desarrollarán a lo largo del cuatrimestre dos prácticas, a elegir entre los cuatro temas genéricos siguientes.

(a) Ampliar (o reimplantar en otro lenguaje) un procesador de lenguaje desarrollado con anterioridad (por ejemplo en Procesadores de Lenguajes) para contemplar algún(os) paradidgmas como orientación a objeto, excepciones, concurrencia, ... (~22,5h)

(b) Retargetting de un procesador (a otra máquina, a bytecode, intérprete, ...) (~22,5h)

(c) Optimización de código en algunas de sus facetas, asimismo de un procesador anterior, normalmente. (~22,5h)

(d) Implementación de paradigma funcional, lógico, distribuido,... (~22,5h)

La documentación final describe la metodología empleada, ejemplos de uso y resultados, así como los código fuente y objeto en formato informático.

Bibliografía

[1 Básico] Compilers: principles, techniques, and tools /

Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman.
Addison-Wesley,, Reading, Mass : (1986)
0201101947

[2 Básico] Parsing techniques :a practical guide /

Dick Grune, Ceriei J.H. Jacobs.
Ellis Horwood,, London : (1990)
0136514316

[3 Básico] Modern compiler design /

Dick Grune, Henri E. Bal, Ceriei J.H. Jacobs, Koen G. Langendoen.
John Wiley & Sons,, Chichester : (2000)
0-471-97697-0

[4 Básico] Engineering a Compiler /

Keith Cooper a Linda Torczon.
Morgan Kaufmann,, San Francisco : (2003)
1-55860-699-8(paperback)

[5 Recomendado] Modern compiler implementation in Java /

Andrew W. Appel with Jens Palsberg.
Cambridge University Press,, Cambridge : (2002) - (2nd ed.)
0-521-82060-X

[6 Recomendado] Programming language pragmatics /

Michael L. Scott.
Morgan Kaufmann,, San Francisco : (2000)
1-55860-578-9

[7 Recomendado] Advanced compiler design and implemetation.

Muchnick, Steven S.

Morgan Kaufmann,, San Francisco (California) : (1997)

1558603204

[8 Recomendado] Compiler design /

Reinhard Wilhelm, Dieter Maurer.

Addison-Wesley,, Wokingham (England) : (1995)

0201422905

Equipo Docente

JOSÉ FORTES GÁLVEZ

(COORDINADOR)

Categoría: *TITULAR DE UNIVERSIDAD*

Departamento: *INFORMÁTICA Y SISTEMAS*

Teléfono: *928458724* **Correo Electrónico:** *jose.fortes@ulpgc.es*