

**▼ Información**

Ficha

Temario

Incompatibilidades

Profesorado

▼ MaterialesLLEU > Ingeniería Informática (Plan de 2001) > **II26 - Procesadores de Lenguaje**

Última actualización: 28/05/2009

Objetivos

- Conocer con detalle la estructura de los procesadores de lenguaje: las partes en que se dividen (análisis léxico, análisis sintáctico, análisis semántico y generación), su interrelación y las principales técnicas utilizadas en cada una de ellas.
- Conocer formalismos para especificar lenguajes de programación.
- Aprender a implementar analizadores descendentes recursivos y a escribir gramáticas adecuadas para dichos analizadores.
- Aprender el interés de dichas técnicas para desarrollar aplicaciones informáticas que deban procesar datos de entrada estructurados (calculadoras, conversores de formato, navegadores web, etc.).
- Poder implementar un compilador o intérprete de un lenguaje de programación con y sin ayuda de un metacompilador.

Evaluación

Se ofrecerán dos itinerarios a los estudiantes.

En el itinerario A la nota final se obtendrá ponderando en un 50% la del examen final y en un 50% el resultado de la evaluación continua de los ejercicios y prácticas realizados por los estudiantes a lo largo del curso dentro de los plazos que se indiquen, pudiéndose exigir una nota mínima en cada parte. La elección del itinerario A deberá quedar reflejada en un contrato de aprendizaje al principio del curso, en el que cada estudiante se comprometa a participar en las actividades del curso y acepte la evaluación continua.

En el itinerario B, pensado fundamentalmente para estudiantes que no puedan asistir regularmente a clase, la evaluación se realizará únicamente mediante el examen final.

En el examen final se permitirá consultar libros y apuntes. Serán materia de examen los conceptos tratados en las clases de teoría, su desarrollo en los materiales de estudio propuestos por el profesor y también las actividades llevadas a cabo en las sesiones de prácticas.

Temario de teoría

- Tema 1. Estructura de los compiladores e intérpretes.
- Tema 2. Análisis léxico.
- Tema 3. Análisis sintáctico.
- Tema 4. Análisis semántico.
- Tema 5. Generación de código.

Bibliografía**Bibliografía básica**

- * Alfred V. Aho, Monica S. Lam, Ravi Sethi i Jeffrey D. Ullman.
"Compilers. Principles, Techniques, and Tools".
Addison Wesley, 2007.
[QA76.76.C65 C66 2007]
([QA76.76.C65 C6618 2008] en castellano)
- * Kenneth C. Louden
"Construcción de compiladores principios y práctica"
México Thomson cop. 2004
[QA76.76.C65 L6818 2004]
- * Keith D. Cooper y Linda Torczon
"Engineering a Compiler"
Elsevier, 2004
[QA76.76.C65 C66 2004]
- * Dick Grune, Henri E. Bal, Cerial J.H. Jacobs y Koen G. Langendoen.
"Modern Compiler Design".
Wiley, 2000.
[QA76.76.C65 M64 2000]

Bibliografía complementaria

- * John R. Levine, Tony Mason y Doug Brown.
"Lex & Yacc".
UNIX Programming Tools.
O'Reilly and Associates, 1992.
[QA76.76.U84 L48 1992]
- * Michael L. Scott
Programming language pragmatics
Morgan Kaufmann cop, 2000.
[QA76.7.S26 2000]

Observaciones

Para poder abordar la asignatura, los estudiantes deben tener conocimientos previos de programación en Python (asignaturas II04 e II24), programación en lenguaje ensamblador (II19), programación orientada a objetos (II17), árboles (II13), entornos de programación (II15) y teoría básica de autómatas y lenguajes formales (II20).

Los estudiantes provenientes de ITIG u otras titulaciones que no hayan cursado dichas asignaturas, y muy especialmente la II20, deben ser conscientes de que la II26 no es una asignatura en la que se enseñan esos contenidos sino en la que se utilizan.