

## **Grado en Ingeniería Informática**

### **30230 - Procesadores de lenguajes**

**Guía docente para el curso 2014 - 2015**

**Curso: 3, Semestre: 2, Créditos: 6.0**

---

## **Información básica**

---

### **Profesores**

- **Francisco Javier Fabra Caro** jfabra@unizar.es

### **Recomendaciones para cursar esta asignatura**

Se recomienda que el alumno haya cursado las siguientes asignaturas:

- Teoría de la Computación (segundo curso, primer semestre)

Además, se hacen las siguientes recomendaciones:

- Estudio de los conceptos teóricos.
- Resolución de los ejercicios planteados en las clases de problemas.
- Realización de las prácticas de laboratorio de forma cuidadosa y durante las fechas recomendadas. Hay una muy estrecha relación entre los conceptos teóricos y el trabajo a desarrollar en las sesiones de laboratorio, por lo que la correcta realización de las prácticas periódicas es una muy buena forma de llegar a los resultados de aprendizaje necesarios para superar la asignatura

### **Actividades y fechas clave de la asignatura**

El calendario detallado de actividades se establecerá a partir del aprobado por la Universidad para el curso académico correspondiente. Las fechas de exámenes y de entrega de trabajos se anunciarán con suficiente antelación durante las clases y en la página del curso (Moodle).

---

## **Inicio**

---

### **Resultados de aprendizaje que definen la asignatura**

**El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...**

**1:**

Comprende el funcionamiento interno de los compiladores y los utiliza eficaz y eficientemente.

**2:**

Aplicar los conocimientos adquiridos sobre gramáticas y autómatas para la especificación de lenguajes y la

construcción de traductores.

**3:**  
Conocer los métodos y técnicas de construcción de traductores.

**4:**  
Dominar el uso de herramientas específicas para la construcción de traductores.

## Introducción

### Breve presentación de la asignatura

El estudiante del Grado en Ingeniería Informática está familiarizado con el uso de intérpretes y compiladores, que lleva manejando intensivamente desde el inicio de sus estudios universitarios. En esta asignatura podrá no sólo entender cómo funcionan esas herramientas de uso común en su día a día, sino que también aprenderá las técnicas y herramientas para su construcción. La comprensión del funcionamiento de estas herramientas le permitirá utilizarlas más eficazmente. Por otro lado, se trata de una asignatura en que confluyen muchos de los conceptos que ha ido aprendiendo a lo largo de los estudios (lenguajes y gramáticas, arquitectura de computadores, estructuras de datos, técnicas de programación, etc.) pero centrados en un problema concreto: la construcción de procesadores de lenguajes.

Las técnicas aprendidas en esta asignatura no deben verse como circunscritas al proceso de construcción de compiladores e intérpretes, sino que son ampliamente utilizadas en otros contextos: editores de textos, formatos de impresión, migración de datos entre bases de datos, procesamiento de información extraída desde páginas Web,... En general, cualquier proceso de traducción en el que la información de entrada pueda describirse formalmente.

---

## Contexto y competencias

---

### Sentido, contexto, relevancia y objetivos generales de la asignatura

#### La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

- Dar a conocer el funcionamiento interno de los compiladores.
- Capacitar al alumno para que aplique los conocimientos adquiridos sobre gramáticas y autómatas para la especificación de lenguajes y la construcción de traductores.
- Dar a conocer al alumno en profundidad los métodos y técnicas de construcción de traductores.
- Capacitar al alumno para utilizar herramientas específicas para la construcción de traductores.
- Lograr que el alumno utilice los traductores y compiladores de forma eficaz y eficiente.

#### Contexto y sentido de la asignatura en la titulación

Procesadores de Lenguajes es una materia obligatoria de la especialidad de Computación que se implanta como una asignatura de 6 créditos ECTS en el segundo semestre del tercer curso del Grado en Ingeniería Informática con un perfil práctico y aplicado.

La ubicación temporal de la asignatura permite enfocar la asignatura de forma muy práctica y aplicada, y que reflexione sobre las características de los compiladores y traductores que ha utilizado o que conoce. Además, los conceptos iniciales relativos al análisis léxico, la construcción de lenguajes y gramáticas y el análisis de los mismos, los ha adquirido previamente a través de la asignatura de Teoría de la Computación, lo que permite actuar de forma directa y práctica, y aplicar los conceptos aprendidos a la construcción de un compilador completo de un lenguaje con un conjunto reducido de tipos de datos e instrucciones.

#### Al superar la asignatura, el estudiante será más competente para...

**1:**  
Conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico,

sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.

**2:**

Evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

## **Importancia de los resultados de aprendizaje que se obtienen en la asignatura:**

El conjunto de los resultados de aprendizaje pueden concretarse indicando que el alumno será capaz de identificar los requisitos de un determinado lenguaje y construir un traductor completo que abarque el análisis léxico, sintáctico y semántico del lenguaje, así como la generación de código correspondiente para su posterior ejecución. El éxito de su aprendizaje resulta de suma importancia para el desarrollo del futuro Ingeniero en Informática, ya que la comprensión de cómo funciona internamente un compilador será fundamental en su día a día cuando trabaje con lenguajes de programación.

---

## **Evaluación**

---

### **Actividades de evaluación**

**El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluación**

**1:**

La evaluación de la asignatura consta de dos partes:

- **Evaluación de trabajo práctico.** Se evaluará la capacidad del alumno para desarrollar un pequeño compilador utilizando las herramientas específicas presentadas. El peso de esta parte de la evaluación será del 40% de la nota final.
- **Examen teórico-práctico.** Se evaluará la correcta adquisición de los conceptos teóricos asociados a la materia, así como la capacidad para la resolución de problemas. El peso de esta parte de la evaluación será del 60% de la nota final. Es obligatorio haber superado previamente la evaluación del trabajo práctico para poder presentarse al examen teórico-práctico.

La calificación final de la asignatura se obtendrá como la media ponderada de ambas partes, teniendo en cuenta que es necesario llegar a obtener, al menos, 5 puntos sobre 10 en cada una de ellas. En caso de no aprobar una de las dos partes, la calificación no podrá ser superior a 4.5 y será igual al mínimo entre la media ponderada de ambas y 4.5 puntos.

La calificación obtenida en cualquiera de las partes superadas en la primera convocatoria tendrá validez para todas las convocatorias del curso.

---

## **Actividades y recursos**

---

### **Presentación metodológica general**

**El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:**

1. La presentación de los contenidos de la asignatura en clases magistrales por parte del profesor.
2. El estudio personal de la asignatura por parte de los alumnos.
3. La resolución de supuestos teórico-prácticos para casos específicos, en sesiones de problemas.
4. El desarrollo de prácticas específicas por parte de los alumnos, guiadas por el profesor, que amplían los conocimientos teóricos y conducentes al desarrollo de un traductor como caso de aplicación real de la asignatura.

## **Actividades de aprendizaje programadas (Se incluye programa)**

**El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades...**

**1:**

Desarrollo del programa de la asignatura en las clases impartidas en el aula.

**2:**

Aplicación de los conceptos y técnicas específicas presentadas en el programa de la asignatura en las sesiones de problemas y prácticas, complementadas con un mecanismo de evaluación continua.

**3:**

Aplicación de los conceptos y técnicas desarrollados durante el curso a través de sesiones tutoradas.

## **Planificación y calendario**

### **Calendario de sesiones presenciales y presentación de trabajos**

El calendario de la asignatura será definido por el centro en el calendario académico del curso correspondiente.

## **Trabajo**

### **Trabajo del estudiante**

#### **Actividades presenciales**

- |  |          |
|--|----------|
| * Actividad de tipo 1 (clases magistrales):  | 30 horas |
| * Actividad de tipo 3 (clases de prácticas): | 30 horas |

#### **Actividades no presenciales:**

- |   |         |
|---|---------|
| * Actividad de evaluación teórico/práctica: | 3 horas |
| * Actividad de evaluación práctica:(*)      | 8 horas |

(\*) sólo para alumnos que no deseen aplicar evaluación continua a la parte práctica

## **Bibliografía**

### **Bibliografía recomendada**

Se recomienda la siguiente bibliografía:

- **Compiladores: principios, técnicas y herramientas.** A. Aho, R. Sethi, J. Ullman. Addison-Wesley Iberoamericana, 1993
- **Advanced Compiler Design and Implementation.** S.S. Muchnick. Morgan Kaufmann Publishers, 1997
- **Programming Language Pragmatics.** M.L. Scott. Morgan Kaufmann Publishers, 2000
- **Engineering a Compiler.** K.D. Cooper & L. Torczon. Morgan Kaufmann Publishers, 2004

## **Programa**

### **Programa de la asignatura**

#### **Sesiones magistrales: 30 horas**

Las actividades de aprendizaje presenciales en el aula, organizadas en clases magistrales y de problemas, se centrarán en el estudio de los siguientes temas:

- Tema 1: Introducción. Traducción y procesamiento de lenguajes.

- Tema 2: Análisis léxico. Autómatas y expresiones regulares. Implementación de un analizador léxico.
- Tema 3: Análisis sintáctico. Gramáticas y clasificación. Transformaciones. Implementación de un analizador semántico.
- Tema 4: Análisis semántico. Tabla de símbolos. Implementación de un analizador semántico.
- Tema 5: Entornos de ejecución.
- Tema 6: Generación y optimización de código.

#### **Sesiones de prácticas de laboratorio y sesiones de problemas: 30 horas**

Se realizarán cinco sesiones de prácticas a través de las que se abordará la construcción de un compilador para un lenguaje sencillo desde cero, así como una máquina virtual que permitirá interpretar y ejecutar dicho lenguaje.

## **Referencias bibliográficas de la bibliografía recomendada**

### **Escuela Politécnica Superior**

- Aho, Alfred V.. Compiladores : principios, técnicas y herramientas / Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman . - 1a ed., 1a reimpr. Wilmington, Delaware : Addison-Wesley Iberoamericana, 1998
- Cooper, K.D. Engineering a Compiler / K.D. Cooper & L. Torczon. Morgan Kaufmann Publishers, 2004
- Muchnick, Steven S.. Advanced compiler design and implementation / Steven S. Muchnick . San Francisco, California : Morgan Kaufmann, cop. 1997
- Scott, Michael Lee. Programming language pragmatics / Michael L. Scott San Francisco [etc.] : Morgan Kaufman, cop. 2000

### **Escuela Universitaria Politécnica**

- Compiladores : principios, técnicas y herramientas / Alfred V. Aho ... [et al.] ; traducción Alfonso Vidal Romero Elizondo ; revisión técnica Sergio Fuenlabrada Velázquez ... [et al.] . 2<sup>a</sup> ed. Naucalpan de Juárez (Edo. de México) : Pearson Educación, 2008
- Cooper, K.D.. Engineering a compiler / K.D. Cooper, L. Torczon. San Francisco (etc.) / Morgan Kaufmann Publishers, 2004
- Muchnick, Steven S.. Advanced compiler design and implementation / Steven S. Muchnick . San Francisco, California : Morgan Kaufmann, cop. 1997
- Scott, Michael Lee. Programming language pragmatics / Michael L. Scott . San Francisco [etc.] : Morgan Kaufman, cop. 2000