

## 30230 - Procesadores de lenguajes

### Información del Plan Docente

**Año académico:** 2019/20

**Asignatura:** 30230 - Procesadores de lenguajes

**Centro académico:** 110 - Escuela de Ingeniería y Arquitectura

**Titulación:** 439 - Graduado en Ingeniería Informática

**Créditos:** 6.0

**Curso:** 3

**Periodo de impartición:** Segundo semestre

**Clase de asignatura:** ---

**Materia:** ---

## 1. Información Básica

### 1.1. Objetivos 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.

### 1.2. 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.

### 1.3. Recomendaciones para cursar la asignatura

Se recomienda que el alumno haya cursado las siguientes asignaturas:

- Teoría de la Computación

Además, se hacen las siguientes recomendaciones:

- Estudio de los conceptos teóricos.
- Resolución de ejercicios relacionados con los conceptos teóricos.
- 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

## 2. Competencias y resultados de aprendizaje

### 2.1. Competencias

**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.

## 2.2.Resultados de aprendizaje

**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.

## 2.3.Importancia de los resultados de aprendizaje

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.

## 3.Evaluación

### 3.1.Tipo de pruebas y su valor sobre la nota final y criterios de evaluación para cada prueba

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

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.

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.

## 4.Metodología, actividades de aprendizaje, programa y recursos

### 4.1.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.

### 4.2.Actividades de aprendizaje

**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.
3. Aplicación de los conceptos y técnicas desarrollados durante el curso a través de sesiones tutorizadas.

### **4.3.Programa**

#### **Programa de la asignatura**

##### **Sesiones magistrales**

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 sesiones de prácticas a través de las que se abordará la construcción de un compilador para un lenguaje sencillo desde cero, resultando en la generación de código y ejecución de dicho lenguaje.

### **4.4.Planificación de las actividades de aprendizaje y calendario de fechas clave**

#### **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 del estudiante**

##### **Actividades presenciales**

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

##### **Actividades no presenciales: 85 horas (aprox.)**

- Actividad de evaluación teórico/práctica: 3 horas
- Actividad de evaluación práctica: 3 horas

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).

### **4.5.Bibliografía y recursos recomendados**

[BB: Bibliografía básica / BC: Bibliografía complementaria]

<http://psfunizar7.unizar.es/br13/egAsignaturas.php?codigo=30230&Identificador=14694>

- [BB] 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
- [BB] Cooper, Keith D. Engineering a compiler / Keith D. Cooper, Linda Torczon . 2nd ed. San Francisco : Morgan Kaufmann, cop. 2012
- [BB] Muchnick, Steven S.. Advanced compiler design and implementation / Steven S. Muchnick . San Francisco, California : Morgan Kaufmann, cop. 1997
- [BB] Scott, Michael Lee. Programming language pragmatics / Michael L. Scott San Francisco [etc.] : Morgan Kaufman, cop. 2000

