

**Guía docente de la asignatura**

<b>Asignatura</b>	LENGUAJES DE PROGRAMACIÓN		
<b>Materia</b>	COMPUTACIÓN		
<b>Módulo</b>			
<b>Titulación</b>	GRADO EN INGENIERÍA INFORMÁTICA (463)		
<b>Plan</b>	463	<b>Código</b>	45202
<b>Periodo de impartición</b>	1 <sup>er</sup> . CUATRIMESTRE	<b>Tipo/Carácter</b>	OBLIGATORIA
<b>Nivel/Ciclo</b>	GRADO	<b>Curso</b>	3º
<b>Créditos ECTS</b>	6 ECTS		
<b>Lengua en que se imparte</b>	CASTELLANO		
<b>Profesor/es responsable/s</b>	VALENTÍN CARDEÑOSO PAYO M <sup>a</sup> LUISA GONZÁLEZ DÍAZ		
<b>Datos de contacto (E-mail, teléfono...)</b>	TELÉFONO: 983 423000 ext. 5601 / ext. 5616 E-MAIL: <a href="mailto:valen@infor.uva.es">valen@infor.uva.es</a> , <a href="mailto:mluisa@infor.uva.es">mluisa@infor.uva.es</a>		
<b>Horario de tutorías</b>	Véase <a href="http://www.uva.es">www.uva.es</a> → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías		
<b>Departamento</b>	DEPARTAMENTO DE INFORMÁTICA (ATC, CCIA, LSI)		

## 1. Situación / Sentido de la Asignatura

---

### 1.1 Contextualización

---

Los lenguajes de programación juegan un papel central en la formación de cualquier graduado en Ingeniería Informática. Después de haber adquirido una cierta competencia en la programación y en la solución algorítmica de problemas, el paso natural es analizar por qué los lenguajes de programación se diseñan tal y como los conocemos.

El estudio de los lenguajes de programación conecta los aspectos teóricos sobre los que se fundamenta la computación con el origen de la solución algorítmica de problemas y con las arquitecturas actuales sobre las que se ejecutan los programas, así como con los aspectos metodológicos relacionados con la producción de software y servicios de calidad.

La estructura y propiedades de las diversas familias de lenguajes de programación guardan además una relación directa con la forma en que se representa y procesa la información usando computadoras y redes de computadoras, lo que constituye el núcleo conceptual y estructural de la disciplina.

Esta asignatura se organiza en torno a la propuesta de unidades de conocimiento propuestas en el Computing Curricula 2001 de la ACM en relación con los Lenguajes de Programación y se orienta más al análisis de la estructura y funcionamiento de los lenguajes de programación que a la revisión del inmenso catálogo de familias y ejemplares de lenguajes existentes. Con ello, se sientan las bases para comprender en mayor profundidad las características y posibilidades que ofrecen los diversos lenguajes de programación que el graduado tenga que afrontar a lo largo de su vida profesional y, al mismo tiempo, se establece un punto de partida para el diseño de lenguajes, lo que sin duda puede tener una importancia capital para el desarrollo de software y servicios en el futuro.

### 1.2 Relación con otras materias

---

En consonancia con todo ello, esta asignatura se ofrece en el tercer curso del grado en Ingeniería Informática, orientado a un perfil profesional de desarrollo de software y servicios de carácter general. Aparece después de las asignaturas de Fundamentos de Programación, Paradigmas de Programación, Interacción Persona Computadora y Fundamentos de Ingeniería del Software y en paralelo con la Programación Orientada a Objetos, materias con las que está directamente relacionada. Sigue también a otras materias relacionadas con la plataforma tecnológica, como los Fundamentos de Computadoras, los Fundamentos de Sistemas Operativos, los Fundamentos de Redes o los Sistemas Distribuidos, cuya superación favorecerá sin duda una mayor facilidad en la consecución de las competencias básicas asociadas a esta asignatura.

### 1.3 Prerrequisitos

---

Aunque no se establezcan como condiciones previas, los alumnos que cursen esta asignatura deberían haber adquirido las competencias básicas de programación asociadas a las materias de Fundamentos de Programación, Paradigmas de Programación y las competencias básicas metodológicas que proporcionan las asignaturas de Fundamentos de Ingeniería del Software y de Interacción Persona Computadora. Se supone también un conocimiento de los aspectos arquitectónicos básicos contenidos en las asignaturas de Fundamentos de Computadores y de Arquitectura y Organización de Computadores.



## 2. Competencias

### 2.1 Generales

Código	Descripción
G03	Capacidad de análisis y síntesis
G04	Capacidad de organizar y planificar
G05	Comunicación oral y escrita en la lengua propia
G06	Conocimiento de una segunda lengua (preferentemente inglés)
G08	Habilidades de gestión de la información
G09	Resolución de problemas
G10	Toma de decisiones
G11	Capacidad crítica y autocrítica
G12	Trabajo en equipo
G14	Responsabilidad y compromiso ético
G15	Liderazgo
G16	Capacidad de aplicar los conocimientos en la práctica
G17	Habilidades de investigación
G18	Capacidad de aprender
G19	Capacidad de adaptarse a nuevas situaciones
G20	Capacidad de generar nuevas ideas
G21	Habilidad para trabajar de forma autónoma

### 2.2 Específicas

Código	Descripción
CC2	Capacidad para 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.
CI8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

## 3. Objetivos

Código	Descripción
CC2.1	Conocer los fundamentos de la teoría de lenguajes formales y su conexión con la programación.
CC2.2	Entender los principios de diseño de los lenguajes de programación y saber cómo relacionar sus componentes estructurales y funcionales básicos.
CC2.3	Ser capaz de construir analizadores léxicos y sintácticos para lenguajes de programación sencillos.
CC2.4	Ser capaz de diseñar analizadores semánticos para lenguajes de programación sencillos.
CI8.1	Saber aplicar el conjunto de restricciones de diseño impuestas por el entorno de programación y ejecución concreto de un lenguaje de programación.
CI8.2	Conocer los principios y técnicas básicas de la optimización local y global de código.

**4. Tabla de dedicación del estudiante a la asignatura**

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	28	Estudio y trabajo autónomo individual	48
Clases prácticas de aula (A)			
Laboratorios (L)	20	Estudio y trabajo autónomo grupal	30
Prácticas externas, clínicas o de campo			
Seminarios (S)	6	Estudio y trabajo autónomo grupal	12
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	6		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>

## 5. Bloques temáticos

### Bloque 1: Fundamentos del Procesamiento de Lenguajes

Carga de trabajo en créditos ECTS: **2,0**

#### a. Contextualización y justificación

En este primer bloque se sientan las bases formales y prácticas de la construcción de procesadores de lenguajes. Se introducirán los aspectos básicos del proceso de compilación e interpretación así como los rudimentos de la teoría de lenguajes formales que sirvan para fundamentar la división del procesamiento de un lenguaje en los diferentes niveles: léxico, sintáctico, semántico y generación de código. En este bloque se describen y analizan los componentes básicos de un traductor y de un intérprete, desde un punto de vista funcional.

En el apartado práctico, se desarrollará una primera actividad que permita al estudiante familiarizarse con entornos de generación de analizadores estándar, como base para desarrollos posteriores. En una segunda entrega se propondrá un estudio de la estructura y bloques funcionales de un compilador abierto de calidad comercial, como es GCC, como vía para comprender los conceptos de fase y paso de compilación, así como para tener un primer contacto con lenguajes de representación intermedia.

#### b. Objetivos de aprendizaje

Código	Descripción
CC2.1	Conocer los fundamentos de la teoría de lenguajes formales y su conexión con la programación.
CC2.3	Ser capaz de construir analizadores léxicos y sintácticos para lenguajes de programación sencillos.
CI8.2	Conocer los principios y técnicas básicas de la optimización local y global de código.

#### c. Contenidos

### TEORÍA

#### TEMA 1: Introducción al procesamiento de lenguajes

- 1.1 Planteamiento del problema.
- 1.2 Espectro de lenguajes de programación.
- 1.3 Compilación e interpretación.
- 1.4 Entornos de programación
- 1.5 Anatomía de un compilador.

#### TEMA 2: Sintaxis de los lenguajes de programación

- 2.1 Especificación formal de la estructura léxica y sintáctica.
- 2.2 Extracción de componentes léxicos
- 2.3 Estrategias de Análisis sintáctico
- 2.4 Bases teóricas del procesado léxico y sintáctico.

#### TEMA 3: Aspectos estructurales globales

- 3.1 Momento de asociación (binding time)
- 3.2 Tiempo de vida y gestión de almacenamiento.
- 3.3 Alcance y visibilidad: concepto y reglas.
- 3.4 Implementación de la asociación en procesamiento de lenguajes.

#### TEMA 4: Análisis Sintáctico

- 4.1 Análisis determinista descendente.
- 4.2 Análisis determinista ascendente.
- 4.3 Árboles de Sintaxis abstracta (AST)

**TEMA 5: Análisis Semántico y Generación de Código**

- 1.1 Gramáticas de atributos.
- 1.2 Decoración y recorrido de AST.
- 1.3 Generación de código.
- 1.4 Aspectos básicos de la optimización de código.

**PRÁCTICAS****P0: Entornos de generación de analizadores****P1: Anatomía del compilador GCC****d. Métodos docentes**

---

- Ver anexo

**e. Plan de trabajo**

---

- Ver anexo.

**f. Evaluación**

---

- Ver el punto 7 de la guía.

**g. Bibliografía básica**

---

- Michael L. Scott, *Programming Language Pragmatics*, 3rd. ed., Morgan Kaufmann, 2009. ISBN 978-0-12-374514-9.
- Terrence W. Pratt y Marvin V. Zelkowitz, *Lenguajes de Programación: Diseño e Implementación*, 3ª. ed., Prentice-Hall, 1998. ISBN 970-17-0046-5.

**h. Bibliografía complementaria**

---

- Alfred Aho, Ravi Sethi y Jeffrey Ullman, *Compiladores: Principios, Técnicas y Herramientas*, 2<sup>nd</sup> ed. Pearson, 2008. ISBN 978-970-26-1133-2.
- Robert Harper, *Practical foundations for Programming Languages*. Creative Commons Licence. 2012.

**i. Recursos necesarios**

---

Todos los recursos necesarios para el desarrollo de las actividades teóricas y prácticas estarán a disposición de los alumnos en la página de la asignatura en el campus virtual.

**Bloque 2: Bases del Diseño de Lenguajes**Carga de trabajo en créditos ECTS: **a. Contextualización y justificación**

En este segundo bloque nos centramos en la estructura de datos y control y en los mecanismos de abstracción más importantes que se soportan en muchos lenguajes de programación. Este es el bloque central de la asignatura, en el que comenzaremos analizando las diferentes variantes de control de flujo presentes en los lenguajes de programación. A continuación nos concentramos en el estudio de los tipos de datos, tanto desde un punto de vista descriptivo como desde un punto de vista algebraico, analizando los sistemas de tipos y los problemas de equivalencia y conversión de tipo. En una tercera parte, se estudiarán los mecanismos de abstracción del control de flujo que dan entrada a las diversas formas de modularización de código, con especial atención a los problemas de paso de parámetros y aspectos avanzados como las excepciones y los eventos. El bloque termina analizando los mecanismos de abstracción sobre datos que dan soporte a la orientación al objeto, desde los mecanismos básicos de encapsulamiento y herencia hasta los aspectos de enlazamiento o asociación dinámica.

En el apartado práctico, se irá desarrollando por fases y entregas separadas la componente léxica sintáctica y semántica de un intérprete sencillo, que se completará con un comprobador de tipos básico.

**b. Objetivos de aprendizaje**

Código	Descripción
CC2.2	Entender los principios de diseño de los lenguajes de programación y saber cómo relacionar sus componentes estructurales y funcionales básicos.
CC2.4	Ser capaz de diseñar analizadores semánticos para lenguajes de programación sencillos.
CI8.1	Saber aplicar el conjunto de restricciones de diseño impuestas por el entorno de programación y ejecución concreto de un lenguaje de programación.

**c. Contenidos****TEORÍA****TEMA 1: Control de flujo**

- 1.1 Evaluación de expresiones
- 1.2 Flujo estructurado y no estructurado
- 1.3 Secuencia
- 1.4 Selección
- 1.5 Iteración
- 1.6 Recursión

**TEMA 2: Tipos de Datos**

- 2.1 Sistemas de tipos
- 2.2 Comprobación de tipos
- 2.3 Registros y variantes
- 2.4 Arreglos, Tiras y Conjuntos
- 2.5 Apuntadores y tipos recursivos.
- 2.6 Comprobación de igualdad y asignación.

**TEMA 3: Abstracción del control de flujo**

- 3.1 Distribución de la pila.
- 3.2 Secuencias de llamadas.
- 3.3 Pasos de parámetros.
- 3.4 Módulos y genéricos.
- 3.5 Gestión de excepciones.
- 3.6 Corutinas.
- 3.7 Eventos.

**TEMA 4: Abstracción de Datos y Orientación a Objetos**

- 4.1 Encapsulamiento y Herencia



- 4.2 Inicialización y Finalización
- 4.3 Asociación dinámica de métodos.

## **PRÁCTICAS**

- P2: Analizador léxico sencillo**
- P3: Analizador sintáctico sencillo**
- P4: Intérprete sencillo**

### **d. Métodos docentes**

---

- Ver anexo

### **e. Plan de trabajo**

---

- Ver anexo

### **f. Evaluación**

---

- Ver el punto 7 de la guía.

### **g. Bibliografía básica**

---

- Michael L. Scott, *Programming Language Pragmatics*, 3rd. ed., Morgan Kaufmann, 2009. ISBN 978-0-12-374514-9.
- Terrence W. Pratt y Marvin V. Zelkowitz, *Lenguajes de Programación: Diseño e Implementación*, 3ª. ed., Prentice-Hall, 1998. ISBN 970-17-0046-5.

### **h. Bibliografía complementaria**

---

- Robert Harper, *Practical foundations for Programming Languages*. Creative Commons Licence. 2012.

### **i. Recursos necesarios**

---

Todos los recursos necesarios para el desarrollo de las actividades teóricas y prácticas estarán a disposición de los alumnos en la página de la asignatura en el campus virtual.



**Bloque 3: Modelos de programación**Carga de trabajo en créditos ECTS: **a. Contextualización y justificación**

En este tercer bloque abordamos una presentación de los principales rasgos que caracterizan a tres familias de lenguajes de programación que no por estar tan extendidos como la familia de lenguajes procedimentales o imperativos tienen menos importancia. Se cubre el paradigma funcional, lógico y los lenguajes de scripting, por su enorme importancia práctica en el escenario actual y en el futuro de la web.

En el apartado práctico, se abordará un estudio práctico comparativo entre varios lenguajes de programación, representantes de otros tantos paradigmas, en los que se deberá elaborar la solución a un problema común desde la perspectiva imperativa, funcional, lógica y de scripting.

**b. Objetivos de aprendizaje**

Código	Descripción
CC2.2	Entender los principios de diseño de los lenguajes de programación y saber cómo relacionar sus componentes estructurales y funcionales básicos.
CI8.1	Saber aplicar el conjunto de restricciones de diseño impuestas por el entorno de programación y ejecución concreto de un lenguaje de programación.

**c. Contenidos****TEORÍA****TEMA 1: Lenguajes Funcionales**

- 1.1 Introducción histórica
- 1.2 Bases conceptuales de la programación funcional.
- 1.3 Orden de evaluación.
- 1.4 Funciones de orden superior (Formas Funcionales)
- 1.5 Referentes: Scheme y ML

**TEMA 2: Lenguajes Lógicos**

- 2.1 Introducción histórica
- 2.2 Bases conceptuales de la programación lógica.
- 2.7 Resolución, Unificación y Mundos cerrados.
- 2.8 Referentes: Prolog

**TEMA 3: Lenguajes de 'scripting'**

- 3.1 Introducción histórica y concepto de lenguaje de scripting.
- 3.2 Dominios de interés para lenguajes de scripting.
- 3.3 Lenguajes de scripting y WWW.
- 3.4 Aspectos innovadores de los lenguajes de scripting.

**PRÁCTICAS****P5: Comparativa de lenguajes de programación****d. Métodos docentes**

- Ver anexo

**e. Plan de trabajo**

- Ver anexo

**f. Evaluación**

- Ver el punto 7 de la guía.

**g. Bibliografía básica**



- Michael L. Scott, *Programming Language Pragmatics*, 3<sup>rd</sup>. ed., Morgan Kaufmann, 2009. ISBN 978-0-12-374514-9.
- Terrence W. Pratt y Marvin V. Zelkowitz, *Lenguajes de Programación: Diseño e Implementación*, 3<sup>a</sup>. Ed., Prentice-Hall, 1998. ISBN 970-17-0046-5.

---

#### **h. Bibliografía complementaria**

- Robert Harper, *Practical foundations for Programming Languages*. Creative Commons Licence. 2012.

---

#### **i. Recursos necesarios**

Todos los recursos necesarios para el desarrollo de las actividades teóricas y prácticas estarán a disposición de los alumnos en la página de la asignatura en el campus virtual.

## 6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
<b>Bloque 1: Fundamentos del Procesamiento de Lenguajes</b>	2,0 ECTS	Semanas 1 a 5
<b>Bloque 2: Bases del Diseño de Lenguajes</b>	2,8 ECTS	Semanas 6 a 12
<b>Bloque 3: Modelos de Programación</b>	1,2 ECTS	Semanas 13 a 15

## 7. Sistema de calificaciones – Tabla resumen

Cada una de las actividades dirigidas que realizará el alumno, será objeto de valoración y será tenida en cuenta en el momento de medir la consecución de los resultados de aprendizaje contenidos en esta asignatura. Se procurará que el alumno tenga a su disposición lo antes posible una realimentación adecuada que le permita dirigir su trabajo y sus actividades de estudio. Entre los instrumentos que se contemplan con este fin se incluyen los siguientes elementos: cuestionarios de teoría, entregas de ejercicios prácticos y presentación pública de resultados en los seminarios, que será incorporada a la calificación de la parte práctica de la asignatura.

Mediante los cuestionarios, el alumnado recibe una realimentación de los objetivos cognitivos conseguidos hasta el momento, sobre preguntas similares a las que se propondrán en el examen de teoría normativo, y su influencia en la nota se presenta como algo atractivo, pero no es imprescindible para conseguir la evaluación final positiva. Las entregas de ejercicios prácticos y su presentación son instrumentos de progreso colaborativo e individual que serán evaluados para constatar los resultados de aprendizaje correspondientes a las habilidades específicas que cubre la asignatura, además de los objetivos transversales de aprendizaje como son la capacidad de planificación, de trabajar en grupo y de comunicación. Es preciso obtener una calificación suficiente en esta parte, para obtener una evaluación final positiva en la convocatoria ordinaria.

Finalmente, se propondrá una prueba de competencias teóricas (Examen de teoría) como viene siendo habitual que habrá que superar satisfactoriamente para superar la asignatura, si bien su peso en la nota final en la convocatoria ordinaria es del 40%.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Cuestionarios Teoría (2 x 1 hora)	15 %	Examen parcial contenidos teóricos. Semana 5 y Semana 13
Entregas ejercicios prácticos (5)	45 %	Trabajo continuo práctico. Semanas 2, 4, 7, 9 y 12
Examen de teoría	40 %	Ejercicios y cuestiones cortas Ver calendario exámenes

### CRITERIOS DE CALIFICACIÓN

**Para superar la asignatura será preciso cumplir todas las condiciones siguientes:**

- Alcanzar al menos un 50% de la calificación máxima de la parte de entregas de ejercicios prácticos.
- Alcanzar al menos un 40% de la calificación máxima del examen de teoría o al menos el 50% de la suma de calificaciones máximas de los cuestionarios y del examen de teoría.
- Obtener una calificación final resultante igual o mayor que 5,0.

**Convocatoria extraordinaria: Deberán cumplirse todas las condiciones siguientes:**

- Superar (al menos el 50% de la máxima calificación) el examen de teoría: 70% de la calificación final.
- Superar (al menos el 50% de la máxima calificación) de un supuesto práctico: 30% de la calificación final. El enunciado del supuesto práctico se publicará una vez finalizada la convocatoria ordinaria.
- Obtener una calificación final resultante igual o mayor que 5,0.

**8. Anexo: Métodos docentes**

Actividad	Metodología
<b>Clase de teoría</b>	<ul style="list-style-type: none"><li>• Clase magistral participativa.</li><li>• Estudio de casos en aula.</li><li>• Resolución de problemas.</li></ul>
<b>Clase práctica</b>	<ul style="list-style-type: none"><li>• Resolución de problemas prácticos en laboratorio.</li><li>• Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos de 3 alumnos siguiendo un enfoque colaborativo.</li></ul>
<b>Seminarios</b>	<ul style="list-style-type: none"><li>• Talleres de presentación de trabajos monográficos y revisión de proceso de aprendizaje.</li></ul>
<b>Tutoría</b>	<ul style="list-style-type: none"><li>• Seguimiento de trabajo del alumno y atención de consultas y dudas.</li></ul>



**9. Anexo: Cronograma de actividades previstas**

Sem	FECHA	T	L	S	E	CONTENIDOS/ACTIVIDADES	ENTREGAS	EVAL (%)	HP	HNP
S1	24/09/12	2	2			Bloque 1: Fundamentos del Procesamiento de Lenguajes			4	6
S2	01/10/12	2	2			Bloque 1: Fundamentos del Procesamiento de Lenguajes	Práctica P0	3%	4	6
S3	08/10/12	2	2			Bloque 1: Fundamentos del Procesamiento de Lenguajes			4	6
S4	15/10/12	2	2			Bloque 1: Fundamentos del Procesamiento de Lenguajes	Práctica P1	7%	4	6
S5	22/10/12	2		2		Bloque 1: Fundamentos del Procesamiento de Lenguajes	Cuestionario C1	5%	4	6
S6	29/10/12	2	2			Bloque 2: Bases del Diseño de Lenguajes			4	6
S7	05/11/12	2	2			Bloque 2: Bases del Diseño de Lenguajes	Práctica P2	10%	4	6
S8	12/11/12	2	2			Bloque 2: Bases del Diseño de Lenguajes			4	5
S9	19/11/12	2	2			Bloque 2: Bases del Diseño de Lenguajes	Práctica P3	10%	4	6
S10	26/11/12	2		2		Bloque 2: Bases del Diseño de Lenguajes			4	4
S11	03/12/12	2	2			Bloque 2: Bases del Diseño de Lenguajes			4	4
S12	10/12/12	2	2			Bloque 2: Bases del Diseño de Lenguajes	Práctica P4	10%	4	6
S13	17/12/12	2	2			Bloque 3: Modelos de Programación	Cuestionario C2	10%	4	6
S14	07/01/13	2	2			Bloque 3: Modelos de Programación			4	6
S15	14/01/13	2		2		Bloque 3: Modelos de Programación	Práctica P5	5%	4	8
S16	21/01/13					<i>ENTREGAS Y EVALUACIÓN FINAL</i>			0	
S17	28/01/13				2	<i>ENTREGAS Y EVALUACIÓN FINAL</i>		Examen	40%	2
S18	04/02/13					<i>ENTREGAS Y EVALUACIÓN FINAL</i>			0	
TOTALES		30	24	6	2			100%	62	87
<b>PRESENCIALES</b>		<b>62</b>		<b>DEDICACIÓN ALUMNO</b>		<b>149</b>				