

**Guía docente de la asignatura**

Asignatura	LENGUAJES DE PROGRAMACIÓN		
Materia	3.1 Ingeniería de Software		
Módulo	3. Tecnologías específicas		
Titulación	GRADO EN INGENIERÍA INFORMÁTICA (545)		
Plan	545	Código	46926
Periodo de impartición	2º CUATRIMESTRE	Tipo/Carácter	OB de Mención Ingeniería de Software, OP*
Nivel/Ciclo	GRADO	Curso	3º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	Mª LUISA GONZÁLEZ DÍAZ		
Datos de contacto (E-mail, teléfono...)	TELÉFONO: 983 423000 ext. 5615 E-MAIL: mluisa@infor.uva.es		
Horario de tutorías	Ver www.uva.es → Grados → Grado en Ingeniería Informática → Tutorías		
Departamento	DEPARTAMENTO DE INFORMÁTICA (ATC, CCIA, LSI)		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Los lenguajes de programación desempeñan 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 (1º), Paradigmas de Programación (1º), Interacción Persona Computadora (2º), Fundamentos de Ingeniería del Software (2º) y Programación Orientada a Objetos (2º), materias con las que está directamente relacionada. Sigue también a otras materias relacionadas con la plataforma tecnológica, como los Fundamentos de Computadoras (1º), los Fundamentos de Sistemas Operativos (2º), los Fundamentos de Redes (1º) o los Sistemas Distribuidos (2º), 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, Programación Orientada a Objetos 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
CG2	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria.
CG3	Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan
CG4	Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación específicas indicadas en la sección 3.1 de la memoria de Grado
CG5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación específicas indicadas en la sección 3.1 de la memoria de Grado
CG6	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de específicas indicadas en la sección 3.1 de la memoria de Grado
CG8	Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
CG9	Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática
CG10	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de específicas indicadas en la sección 3.1 de la memoria de Grado

2.2 Transversales

Código	Descripción
CT1	Capacidad de análisis y síntesis
CT2	Capacidad de organizar y planificar
CT3	Comunicación oral y escrita en la lengua propia
CT4	Capacidad para la lectura de textos técnicos en inglés
CT5	Habilidades de gestión de la información
CT6	Resolución de problemas
CT7	Toma de decisiones
CT8	Capacidad crítica y autocrítica
CT9	Trabajo en equipo
CT10	Capacidad de trabajar en un equipo interdisciplinar
CT11	Responsabilidad y compromiso ético
CT12	Liderazgo
CT13	Capacidad de aplicar los conocimientos en la práctica
CT14	Capacidad de aprender
CT15	Capacidad de adaptarse a nuevas situaciones
CT16	Habilidad para trabajar de forma autónoma
CT17	Iniciativa y espíritu emprendedor

2.3 Específicas Comunes a la Rama de Informática

Código	Descripción
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.

2.4 Específicas de la Mención de Ingeniería de Software

Código	Descripción
IS4	Capacidad para identificar y analizar problemas y diseñar, construir, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

3. Objetivos

Código	Descripción
	Conocer los fundamentos de la teoría de lenguajes formales y su conexión con la programación.
	Entender los principios de diseño de los lenguajes de programación y saber cómo relacionar sus componentes estructurales y funcionales básicos.
	Ser capaz de construir analizadores léxicos y sintácticos para lenguajes de programación sencillos.
	Ser capaz de diseñar analizadores semánticos para lenguajes de programación sencillos.
	Saber aplicar el conjunto de restricciones de diseño impuestas por el entorno de programación y ejecución concreto de un lenguaje de programación.
	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)	$30 \cdot 3 \cdot 1 = 27$	Estudio y trabajo autónomo individual	48
Clases prácticas de aula (A)			
Laboratorios (L)	24	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)	$3 \cdot 1 = 3$		
Total presencial	60	Total no presencial	90

5. Bloques temáticos

Bloque 1: Fundamentos del Procesamiento de Lenguajes

Carga de trabajo en créditos ECTS: **4,0**

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 desarrollarán dos primeras actividades para familiarizar al estudiante con las expresiones regulares y el concepto de compilación y sus fases. A continuación se presentarán, en sucesivas sesiones, entornos de generación de analizadores estándar como base para desarrollos posteriores. A continuación se realizará en equipo y gradualmente un traductor sencillo.

b. Objetivos de aprendizaje

Código	Descripción
	Conocer los fundamentos de la teoría de lenguajes formales y su conexión con la programación.
	Ser capaz de construir analizadores léxicos y sintácticos para lenguajes de programación sencillos.

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: Léxico y Sintaxis de los lenguajes de programación

- 2.1 Especificación de la estructura léxica y sintáctica.
- 2.2 Análisis léxico
- 2.3 Bases teóricas del procesado léxico y sintáctico.
- 2.4 Análisis Sintáctico
 - 2.4.1 Análisis descendente
 - 2.4.2 Análisis ascendente
- 2.5 Árboles de Sintaxis Abstracta (AST)

TEMA 4: Análisis Semántico y Generación de Código

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

**PRÁCTICAS**

- P1: Expresiones regulares (individual, con preparación grupal)
P2: Compilación (individual)
P3a: Traductor (grupal, defensa individual o exposición). Fase léxica

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.
- Alfred Aho, Monica S. Lam, Ravi Sethi y Jeffrey Ullman, *Compilers. Principles, Techniques and Tools*, 2nd ed. Pearson, 2007. ISBN 0-321-48681-1.
- Terrence Parr, *The Definitive ANTLR4 Reference*. The Pragmatic Bookshelf, 2012, ISBN-13:978-1-93435-699-9

h. Bibliografía complementaria

- Terrence W. Pratt y Marvin V. Zelkowitz, *Lenguajes de Programación: Diseño e Implementación*, 3^a. ed., Prentice-Hall, 1998. ISBN 970-17-0046-5.
- Robert Harper, *Practical foundations for Programming Languages*. Creative Commons Licence. 2012
- Alfred Aho, Ravi Sethi y Jeffrey Ullman, *Compiladores: Principios, Técnicas y Herramientas*, 2nd ed. Pearson, 2008. ISBN 978-970-26-1133-2..

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 LenguajesCarga de trabajo en créditos ECTS: **a. Contextualización y justificación**

En este segundo bloque enfrentamos las cuestiones cruciales en diseño de lenguajes. Comenzaremos analizando el concepto de “nombre” como abstracción en este contexto y las nociones relacionadas con ello. A continuación 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. Se analizan las diferentes variantes de control de flujo presentes en los lenguajes de programación y 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.

En el apartado práctico, se implantarán, en el traductor que se construye con las técnicas del bloque anterior, los conceptos estudiados en este, incorporando tipos y estructuras de control.

b. Objetivos de aprendizaje

Código	Descripción
	Entender los principios de diseño de los lenguajes de programación y saber cómo relacionar sus componentes estructurales y funcionales básicos.
	Ser capaz de diseñar analizadores semánticos para lenguajes de programación sencillos.
	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 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 5: Control de flujo

- 1.1 Evaluación de expresiones
- 1.2 Flujo estructurado y no estructurado
- 1.3 Estructuras de control de flujo

TEMA 6: Tipos de Datos

- 2.1 Sistemas de tipos
- 2.2 Comprobación de tipos
- 2.3 Comprobación de igualdad y asignación.

PRÁCTICAS

P3b: Traductor (grupal, defensa individual o exposición). Fase sintáctica

P3c: Traductor (grupal, defensa individual o exposición). Fase semántica y traducción

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.
- Alfred Aho, Monica S. Lam, Ravi Sethi y Jeffrey Ullman, *Compilers. Principles, Techniques and Tools*, 2nd ed. Pearson, 2007. ISBN 0-321-48681-1.
- Terrence W. Pratt y Marvin V. Zelkowitz, *Lenguajes de Programación: Diseño e Implementación*, 3^a. ed., Prentice-Hall, 1998. ISBN 970-17-0046-5.
- Terrence Parr, *The Definitive ANTLR4 Reference*. The Pragmatic Bookshelf, 2012, ISBN-13:978-1-93435-699-9

h. Bibliografía complementaria

- Alfred Aho, Ravi Sethi y Jeffrey Ullman, *Compiladores: Principios, Técnicas y Herramientas*, 2nd 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.



6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Fundamentos del Procesamiento de Lenguajes	4,0 ECTS	Primeras 10 semanas
Bloque 2: Bases del Diseño de Lenguajes	2,0 ECTS	Últimas 5 semanas

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, 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 y prácticas (Examen escrito) como viene siendo habitual que habrá que superar satisfactoriamente para superar la asignatura, con un peso en la nota final en la convocatoria ordinaria del 50%.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Cuestionarios Teoría (3 x 1 hora)	15 %	Examen parcial contenidos teóricos. Semanas 5, 10 y 15, aproximadamente
Entregas ejercicios prácticos (5 entregas)	35 %	Trabajo continuo práctico. Entregas y defensas semanas 2, 7, 11, 14 y 15 aproximadamente
Examen escrito	50 %	Ejercicios y cuestiones. Ver calendario exámenes

CRITERIOS DE CALIFICACIÓN

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

- Alcanzar al menos un 45% de la calificación máxima de la parte de entregas de ejercicios prácticos
- Alcanzar al menos un 40% de la calificación máxima del examen escrito o al menos el 50% de la suma de calificaciones máximas de los cuestionarios y del examen escrito.
- Obtener una calificación final resultante igual o mayor que 5,0 con las ponderaciones de la tabla anterior

Convocatoria extraordinaria: Deberán cumplirse todas las condiciones siguientes:

- Superar al menos el 45% de la máxima calificación del examen escrito de segunda convocatoria: 70% de la calificación final.
- Superar al menos el 45% de la máxima calificación de una entrega práctica: 30% de la calificación final. El enunciado de la entrega práctica se comunicará finalizada la convocatoria ordinaria, tendrá relación con las realizadas durante el curso (ampliación, mejora ...) y su calificación tendrá en cuenta la obtenida en estas.
- Obtener una calificación final resultante igual o mayor que 5,0.

**8. Anexo: Métodos docentes**

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none">• Clase magistral participativa.• Estudio de casos en aula.• Resolución de problemas.
Clase práctica	<ul style="list-style-type: none">• Resolución de problemas prácticos en laboratorio.• 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.
Seminarios	<ul style="list-style-type: none">• Talleres de presentación de trabajos monográficos y revisión de proceso de aprendizaje.
Tutoría	<ul style="list-style-type: none">• Seguimiento de trabajo del alumno y atención de consultas y dudas.





9. Anexo: Cronograma de actividades previstas

Aparecerá en el aula virtual de la asignatura, en aulas.inf.uva.es

