



FICHA IDENTIFICATIVA

DATOS DE LA ASIGNATURA

Código: 34672

Nombre: Lenguajes de programación

Ciclo: Grado

Créditos ECTS: 6

Curso académico: 2025-26

TITULACIONES

Titulación	Centro	Curso	Periodo
1400 - Grado en Ingeniería Informática	Escola Tècnica Superior d'Enginyeria	3	Primer cuatrimestre
1407 - Grado en Ingeniería Multimedia	Escola Tècnica Superior d'Enginyeria	4	Primer cuatrimestre

MATERIAS

Titulación	Materia	Carácter
1400 - Grado en Ingeniería Informática	Programación y Computación	OBLIGATORIA
1407 - Grado en Ingeniería Multimedia	Optatividad	OPTATIVA

COORDINACIÓN

ADSUARA FUSTER JOSE ENRIQUE

AREVALILLO HERRAEZ MIGUEL

RESUMEN

"Lenguajes de Programación" es una asignatura obligatoria de tercer curso del Grado en Ingeniería Informática, de 6 créditos ECTS. La asignatura forma parte de la materia "Programación y Computación"

Se parte de los conocimientos de programación ya adquiridos y se profundiza en aspectos avanzados, abordando el proceso de compilación y aspectos clave del paradigma de la programación funcional.

Se propone un enfoque integrador, en el que se utilizan y amplían conceptos ya aprendidos en otras asignaturas de la titulación. Enlazando con la asignatura "Autómatas, Lenguajes Formales y Aplicaciones", se introduce el proceso de compilación, utilizando herramientas generadoras de analizadores léxicos y sintácticos. La asignatura también presenta el paradigma de la programación funcional, desconocido hasta el momento. El paradigma se estudia desde el punto de vista de la aplicación de conceptos como la inmutabilidad y las funciones de orden superior en la escritura de programas.



CONOCIMIENTOS PREVIOS

RELACIÓN CON OTRAS ASIGNATURAS DE LA MISMA TITULACIÓN

No se han especificado restricciones de matrícula con otras asignaturas del plan de estudios.

OTROS TIPOS DE REQUISITOS

Para cursar esta asignatura es recomendable dominar los conceptos básicos de programación, además de haber superado las asignaturas de la misma materia Estructuras de datos y algoritmos, y Autómatas, lenguajes formales y aplicaciones; las asignaturas de Organización de Computadores; y la materia Informática. En particular, los contenidos sobre gramáticas y lenguajes formales estudiados en Autómatas, lenguajes formales y aplicaciones son esenciales para la comprensión de los conceptos relacionados con el proceso de la compilación.

COMPETENCIAS / RESULTADOS DE APRENDIZAJE

1400 - Grado en Ingeniería Informática

G4 - 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 las competencias específicas establecidas.

G5 - Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según las competencias específicas establecidas.

G9 - 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.

R6 - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

R7 - Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

R8 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

DESCRIPCIÓN DE CONTENIDOS



1. Introducción.

Evolución histórica de los lenguajes de programación.
Características de los lenguajes de programación.
Clasificación de los lenguajes de programación.

2. Procesadores de lenguajes.

Tipos: compiladores, intérpretes.
Sintaxis: criterios, elementos sintácticos.
Análisis del programa fuente:
- Análisis léxico.
- Análisis sintáctico (descendente y ascendente).
- Análisis semántico.
Generación del programa objeto:
- Generación de código intermedio.
- Optimización.
Herramientas.

3. Programación Funcional.

Por qué la programación funcional.
Qué es la programación funcional.
Aplicación de conceptos del paradigma funcional.

VOLUMEN DE TRABAJO (HORAS)

ACTIVIDADES PRESENCIALES

Actividad	Horas
Teoría	30,00
Prácticas en aula	10,00
Laboratorio	20,00
Total horas	60,00

ACTIVIDADES NO PRESENCIALES

Actividad	Horas
Asistencia a otras actividades	0,00
Elaboración de trabajos individuales o en grupo	0,00



Estudio y trabajo autónomo	60,00
Preparación de clases	20,00
Preparación de actividades de evaluación	10,00
Resolución de casos prácticos	0,00
Total horas	90,00

METODOLOGÍA DOCENTE

La asignatura incluye actividades presenciales tanto teóricas como prácticas. En las actividades teóricas de carácter presencial se desarrollarán los temas de la asignatura proporcionando una visión global e integradora, analizando con mayor detalle los aspectos clave y de mayor complejidad, fomentando, en todo momento, la participación del alumnado. Estas actividades se complementan con actividades prácticas con el objetivo de aplicar los conceptos básicos y ampliarlos con el conocimiento y la experiencia que se vayan adquiriendo durante la realización de los trabajos propuestos. Comprenden los siguientes tipos de actividades presenciales:

- Clases de problemas y cuestiones en el aula
- Sesiones de discusión y resolución de problemas y ejercicios previamente trabajados por el alumnado
- Prácticas de laboratorio
- Realización de cuestionarios individuales de evaluación en el aula con la presencia del profesorado.

Además de las actividades presenciales, los estudiantes deberán realizar fuera del aula una serie de tareas, tales como trabajos monográficos, búsqueda bibliográfica dirigida, cuestiones y problemas, y la preparación de clases y exámenes. Se utilizará la plataforma de e-learning (Aula Virtual) de la Universitat de València como soporte de comunicación con el alumnado. A través de ella se tendrá acceso al material didáctico utilizado en clase, así como los problemas y ejercicios a resolver.

EVALUACIÓN

En esta asignatura, la evaluación se llevará a cabo mediante evaluación continua, que incluirá tres pruebas escritas referentes a los contenidos teóricos, prácticos y de problemas (P1, P2 y P3). Estas pruebas se realizarán generalmente durante la primera mitad del cuatrimestre (P1), durante la segunda mitad del cuatrimestre (P2) y fuera del horario lectivo en el período de exámenes (P3). Además, la evaluación de las prácticas se hará a partir de un proyecto (Pr). Las pruebas no eliminarán materia y tendrán, por tanto, una ponderación creciente en la nota final. Para aprobar la asignatura será necesario obtener al menos una calificación de 5 en la parte de exámenes, computada como $Pe = 0,15 \cdot P1 + 0,35 \cdot P2 + 0,5 \cdot P3$ y una calificación de al menos 4 en el proyecto de la asignatura (Pr). La calificación final (C) se obtendrá como $C = 0,7Pe + 0,3 \cdot Pr$.

En segunda convocatoria, la calificación será la obtenida en la prueba realizada en el período de exámenes correspondiente. Se mantiene la obligación de entregar el proyecto final (que podrá volverse a entregar hasta el día del examen), requiriendo al menos una calificación de 5.



Todas las pruebas podrán contener partes o secciones de carácter eliminatorio.

En todos los casos (primera y segunda convocatoria), el alumno deberá entregar todas las prácticas y actividades complementarias correctamente resueltas para aprobar la asignatura, dentro del plazo y en la forma adecuada. En caso contrario, la nota máxima obtenible será de 4.

El uso de herramientas de generación automática de código o de respuestas basadas en modelos de lenguaje / herramientas de inteligencia artificial deberá ser debidamente declarado y justificado por el estudiante, y está expresamente prohibido en los exámenes. El uso no declarado de estas herramientas tendrá la consideración de plagio.

En cualquier caso, la evaluación de la asignatura se hará de acuerdo con el Reglamento de evaluación y calificación de la Universitat de València para los títulos de grado y master aprobado por Consejo de Gobierno de 30 de mayo de 2017 (ACGUV 108/2017). Asimismo, la copia, plagio manifiesto o cualquier otra práctica fraudulenta en cualquier actividad que forma parte de la evaluación supondrá la imposibilidad de superar la asignatura, sometiéndose seguidamente a los procedimientos disciplinarios oportunos indicados en el *PROTOCOLO DE ACTUACIÓN ANTE PRÁCTICAS FRAUDULENTAS EN LA UNIVERSITAT DE VALÈNCIA* ([ACGUV 123/2020](#)).

BIBLIOGRAFÍA

- Lenguajes de programación. Principios y paradigmas, Allen B. Tucker, Robert Noonan, McGraw-Hill/Interamericana de España, S.A.U., 2003
- Java a tope: Compiladores, Sergio Gálvez Rojas y Miguel Ángel Mora Mata, Universidad de Málaga, ISBN-13: 9788468910376
Disponibile en Web: <http://www.lcc.uma.es/~galvez/ftp/libros/Compiladores.pdf>
- On LISP, Advanced Techniques for Common LISP, Paul Graham, Prentice Hall, 1993, ISBN 0130305529. Versión gratuita en <http://paulgraham.com/onlisp.html>