GUÍA DOCENTE

ALGORÍTMICA Y PROGRAMACIÓN

Coordinación: MARTÍNEZ RODRÍGUEZ, SANTIAGO

Año académico 2019-20

ALGORÍTMICA Y PROGRAMACIÓN 2019-20

Información general de la asignatura

Denominación	ALGORÍTMICA Y PROGRAMACIÓN
Código	102364
Semestre de impartición	1R Q(SEMESTRE) EVALUACIÓN CONTINUADA
Carácter	

Grado/Máster	Curso	Carácter	Modalidad
Grado en Técnicas de Interacción Digital y de Computación	1	TRONCAL	Presencial

Número de créditos de la asignatura (ECTS)	6
Tipo de actividad, créditos y grupos	

Tipo de actividad	PRALAB		TEORIA
Número de créditos	3	3	3
Número de grupos	1	2	1

Coordinación	MARTÍNEZ RODRÍGUEZ, SANTIAGO
Departamento/s	INFORMATICA E INGENIERIA INDUSTRIAL
Distribución carga docente entre la clase presencial y el trabajo autónomo del estudiante	6 ECTS = 25x6 = 150 horas de trabajo: 40% -> 60 horas presenciales, 60% -> 90 horas de trabajo autónomo del estudiante.
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.
Idioma/es de impartición	Catalán.
Distribución de créditos	Santi Martínez Rodríguez: 3 Ester Marsal Roca: 3

ALGORÍTMICA Y PROGRAMACIÓN 2019-20

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
MARSAL ROCA, ESTER	ester.marsal@udl.cat	3	
MARTÍNEZ RODRÍGUEZ, SANTIAGO	santi.martinez@udl.cat	3	Concertar con el profesor

Información complementaria de la asignatura

Para abordar la asignatura es recomendable mostrar interés en plantear y analizar problemas reales, para los que se deberá buscar y desarrollar una solución tecnológica que los resuelva.
Es también recomendable mostrar capacidad de análisis, de razonamiento lógico, trabajo metódico y organizado, y capacidad crítica.

Los conocimientos y competencias adquiridas en esta asignatura serán de utilidad en asignaturas posteriores con contenidos de programación, estructuras de datos y algorítmica.

Objetivos académicos de la asignatura

Los objetivos de aprendizaje de la asignatura son analizar y diseñar algoritmos para, posteriormente, implementar en un lenguaje de programación compilable.
Concretamente, el lenguaje imperativo escogido es ANSI C/C++ y los problemas a resolver son principalmente de tratamiento y búsqueda en secuencias.

En concreto, los objetivos de aprendizaje del estudiante son los siguientes:

- Diseñar e implementar estructuras algorítmicas adecuadas para resolver las diferentes tipologías de problemas.
- Diseñar e implementar estructuras de datos adecuadas para representar la información propia de cada problema.
- Diseñar e implementar algoritmos iterativos.
- Identificar la tipología del problema y aplicar la estrategia algorítmica adecuada.
- Diseñar e implementar algoritmos para resolver problemas complejos de forma estructurada.
- Diseñar e implementar soluciones algorítmicas básicas utilizando la técnica de diseño descendente.
- Utilizar un entorno de desarrollo de programas basado en un lenguaje de programación de alto nivel.

Competencias

Competencias Básicas

- **B01**. Que los estudiantes hayan demostrado poseer y comprender conocimientos en su área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

Competencias Transversales

- **CT3**. Adquirir capacitación en el uso de las nuevas tecnologías y de las tecnologías de la información y la comunicación.
- **CT5**. Adquirir nociones esenciales del pensamiento científico.

Competencias Generales

- **CG2**. Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas informáticos.
- **CG3**. Capacidad para utilizar plataformas hardware y software adecuadas para el desarrollo y la ejecución de aplicaciones digitales interactivas.
- **CG5**. 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.
- **CG7**. Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad.
- **CG8**. Capacidad para la abstracción y el razonamiento crítico, lógico y matemático.

Competencias Específicas

- **CE2**. Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la computación.
- **CE3**. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos y bases de datos útiles para el desarrollo de aplicaciones informáticas interactivas.
- **CE4**. Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.
- **CE16**. Capacidad para diseñar y evaluar interfaces persona computador que garanticen la usabilidad de los sistemas, servicios y aplicaciones informáticas.
- **CE17**. Saber aplicar los conocimientos de diseño suficientes para proponer y defender un concepto de diseño de un entorno interactivo y desarrollarlo hasta que pueda ser llevado a la práctica utilizando las tecnologías creativas adecuadas a cada proyecto.
- **CE24**. Ser capaz de comprender los factores humanos que intervienen en todo proceso de interacción entre personas y tecnología, así como saber aplicarlos de forma adecuada al diseño de productos y servicios interactivos y sus interfaces.

Contenidos fundamentales de la asignatura

Introducción: Procesos, algoritmos y programas.

Tema 1. Estructuras algorítmicas básicas

1.1 Constantes, variables, tipos elementales, y expresiones válidas
1.2 La asignación, la composición secuencial, la composición alternativa y la composición iterativa
1.3 Entorno de programación

Tema 2. Diseño de programas iterativos

2.1 Acceso secuencial

- Esquemas algorítmicos de tratamiento de secuencias
- Esquemas algorítmicos de búsqueda en secuencias

2.2 Acceso directo. Las tablas

- Tratamiento secuencial de tablas
- Tratamiento directo de tablas
- Tablas unidimensionales y multidimensionales
- Algoritmos de ordenación clásicos

Tema 3. Tratamiento de datos no elementales

3.1. Acciones y funciones
3.2 Mecanismos de paso de parámetros
3.3 Diseño descendente de algoritmos

Ejes metodológicos de la asignatura

Cada semana el estudiante asiste a 2 horas presenciales con Grupo Grande y 2 horas presenciales con Grupo Medio. Las sesiones con Grupo Medio son de prácticas.

Grupo Grande: Clases Teoría y Problemas (3 créditos)

- Parte teórica: clases soportadas con transparencias y/o apuntes.
- Parte de aplicación práctica: se trabaja siempre con problemas y ejercicios de programación. Se dispone de una **colección de problemas**, de los que se van proporcionando soluciones a lo largo del cuatrimestre.

Grupo Medio: Clases Prácticas (3 créditos)

- Clases dirigidas y seguimiento personalizado para grupos de prácticas. Se dispone de una **colección de problemas**, de los que se van proporcionando soluciones a lo largo del cuatrimestre.
- Uso de los compiladores y herramientas de edición.
- Trabajo continuado alrededor del enunciado de las **dos prácticas obligatorias**.

Trabajo Autónomo (no presencial)**:**

- La práctica se completará en horas no presenciales. En las sesiones de Grupo Medio se da apoyo a las prácticas obligatorias que debe desarrollar el estudiante a lo largo de la asignatura de forma autónoma.
- Se recomienda que el alumno resuelva por cuenta propia los problemas de la colección de problemas, a fin de practicar y obtener *feedback* por parte del profesor.

Plan de desarrollo de la asignatura

Sem.	Descripción	Actividad Presencial GG	Actividad Presencial GM	Trabajo autónomo
1	Presentación Introducción	Presentación asignatura. Introducción: procesos, algoritmos y programas.	Utilització de un entorno de programación.	Resolución de ejercicios.
2	Estructuras algorítmicas básicas	T1: Constantes, variables, tipos elementales, expresiones válidas y gestión de la entrada y salida estándar.	Ejercicios de programación.	Resolución de ejercicios.
3	Estructuras algorítmicas básicas	T1: La asignación, la composición secuencial y la composició alternativa.	Ejercicios de programación.	Resolución de ejercicios.
4	Estructuras algorítmicas básicas	T1: La composición iterativa.	Ejercicios de programación.	Resolución de ejercicios.
5	Diseño de programas iterativos	T2: Tratamiento secuencial.	Práctica 1: Descripción de la primera práctica obligatoria.	Resolución de la Práctica 1 en grupos.
6	Diseño de programas iterativos	T2: Búsqueda en secuencias.	Ejercicios de programación. Soporte a la Práctica 1.	Resolución de ejercicios. Resolución de la Práctica 1 en grupos.
7	Diseño de programas iterativos	T2: Aceso directo. El tipo tabla.	Ejercicios de programación. Soporte a la Práctica 1.	Resolución de ejercicios. Resolución de la Práctica 1 en grupos.
8	Diseño de programas iterativos	T2: Ejercicios de acceso directo con tablas: tratamiento y búsqueda.	Ejercicios de programación. Soporte a la Práctica 1.	Resolución de ejercicios. Resolución de la Práctica 1 en grupos.
9		1.er Examen Parcial	Entrega de la Práctica 1.	Estudiar. Resolución de la Práctica 1 en grupos.
10	Diseño de programas iterativos	T2: Tablas multidimensionales.	Algoritmos de ordenación clásicos.	Resolución de ejercicios.
11	Tratamiento de datos no elementales	T3: Acciones y funciones.	Ejercicios de programación.	Resolución de ejercicios.
12	Tratamiento de datos no elementales	T3: Mecanismos de paso de parámetros.	Práctica 2: Descripción de la segunda práctica obligatoria.	Resolución de la Práctica 2 en grupos.
13	Tratamiento de datos no elementales	T3: Diseño descendente de algoritmos.	Ejercicios de programación. Soporte a la Práctica 2.	Resolución de ejercicios. Resolución de la Práctica 2 en grupos.
14	Tratamiento de datos no elementales	T3: Ejercicios de diseño descendente.	Ejercicios de programación. Soporte a la Práctica 2.	Resolución de ejercicios. Resolución de la Práctica 2 en grupos.
15	Tratamiento de datos no elementales	T3: Ejercicios de diseño descendente.	Ejercicios de programación. Soporte a la Práctica 2.	Resolución de ejercicios. Resolución de la Práctica 2 en grupos.
16		2.º Examen Parcial		Estudiar. Resolución de la Práctica 2 en grupos.

Sem.	Descripción	Actividad Presencial GG	Actividad Presencial GM	Trabajo autónomo
17		2.º Examen Parcial	Entrega de la Práctica 2.	Estudiar. Resolución de la Práctica 2 en grupos.
18				
19		Recuperación	Recuperación de las prácticas.	Estudiar. Resolución de las prácticas.

Sistema de evaluación

Actividades de evaluación

Acrónimo	Actividad de Evaluación	Peso	Nota Mínima	En grupo	Obligatoria	Recuperable
EP1	1.er Examen Parcial	25%	4	No	Sí	Sí
EP2	2.º Examen Parcial	35%	4	No	Sí	Sí
PR1	Práctica 1	15%	4	Sí (≤ 2)	Sí	Sí
PR2	Práctica 2	25%	4	Sí (≤ 2)	Sí	Sí
Para aprobar la asignatura es necesario obtener una nota mínima de 4 tanto en las pruebas escritas como en las prácticas. Además, la nota final deberá ser ≥ 5.						
Nota Final = 0,25 · EP1 + 0,35 · EP2 + 0,15 · PR1 + 0,25 · PR2						

Observaciones:

- Si la nota obtenida en la prueba escrita EP2 es ≥ 4, entonces la nota obtenida podrá actuar como recuperación/mejora de la primera prueba escrita EP1, el peso de la cual es del 25%.
- Si la nota obtenida en la prueba escrita EP2 es < 4, entonces el estudiante puede optar a mejorar/recuperar el 60% que representan las pruebas escritas. La recuperación será una única prueba escrita que será evaluada sobre 10 puntos. La nota obtenida sustituirá la nota de las dos pruebas escritas del curso. La nota de la prueba representará el 60% de la nota final y para aprobar la asignatura la nota de la prueba de recuperación deberá ser ≥ 4.
- Si la nota obtenida en la práctica PR2 es ≥ 4, entonces la nota obtenida podrá actuar como mejora/recuperación de la primera práctica PR1, el peso de la cual es del 15%.
- En caso que la práctica PR2 no se supere (nota < 4), la práctica podrá ser recuperada en el periodo de recuperación.

Bibliografía y recursos de información

Algorítmica

- T. Alsinet, J. Argelich and S. Vila. Programació I: Notes de curs. Eines, Edicions de la Universitat de Lleida, en prensa.
- J. Castro, F. Cucker, X. Messeguer, A. Rubio, L. Solano and B. Valles. Curs de Programació. McGraw-Hill, 1992.
- G. Brassard and P. Bratley. Fundamentos de Algoritmia. Prentice Hall, 1997.
- L. Joyanes. Fundamentos de Programación. Algoritmos, Estructuras de Datos y Objetos. McGraw-Hill, 2003.

El lenguaje ANSI C y C++

- H.M. Deitel and P.J. Deitel. Cómo Programar en C/C++. Prentice-Hall, segunda edición, 2002.
- B. Stroustrup. Programming -- Principles and Practice Using C++. Addison Wesley, 2008.
- L. Joyanes. Programación en C++. McGraw-Hill, 2006.