



Universitat de Lleida

GUÍA DOCENTE **TÉCNICAS DE COMPUTACIÓN**

Coordinación: OJEDA CONTRERAS, JESUS

Año académico 2022-23

Información general de la asignatura

Denominación	TÉCNICAS DE COMPUTACIÓN			
Código	102376			
Semestre de impartición	1R Q(SEMESTRE) EVALUACIÓN CONTINUADA			
Carácter	Grado/Máster	Curso	Carácter	Modalidad
	Grado en Técnicas de Interacción Digital y de Computación	2	OBLIGATORIA	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	
	Número de grupos	1	1	
Coordinación	OJEDA CONTRERAS, JESUS			
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 trabajo autónomo del estudiante			
Información importante sobre tratamiento de datos	Consulte este enlace para obtener más información.			
Idioma/es de impartición	Castellano / Catalán			

Profesor/a (es/as)	Dirección electrónica\nprofesor/a (es/as)	Créditos impartidos por el profesorado	Horario de tutoría/lugar
OJEDA CONTRERAS, JESUS	jesus.ojedacontreras@udl.cat	6	

Información complementaria de la asignatura

Para cualquier duda y/o cuestión, podéis enviar un correo electrónico al profesor de la asignatura.

Objetivos académicos de la asignatura

- Ser capaces de caracterizar formalmente los problemas.
- Analizar la eficiencia de los algoritmos mediante el uso de la notación asintótica.
- Identificar la tipología del problema, así como la estrategia algorítmica más adecuada para su resolución.
- Diseñar e implementar estructuras de datos adecuadas para representar la información propia de cada problema.
- Diseñar e implementar estrategias algorítmicas eficientes para resolver las diferentes tipologías de problemas.

Competencias

- CT3. Implementar nuevas tecnologías y tecnologías de la información y la comunicación.
- CG10. Capacidad para aplicar las técnicas algorítmicas adecuadas para la resolución de problemas computacionales.
- 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.
- CE8. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas computacionales, analizando la idoneidad y complejidad de los algoritmos propuestos.
- CE9. Conocer, diseñar y utilizar de forma eficiente los tipos y estructuras de datos más adecuados la resolución de un problema.
- CE10. Capacidad para analizar, diseñar, construir y mantener aplicaciones digitales interactivas de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Contenidos fundamentales de la asignatura

1. Preliminares: Notación y definiciones
2. Especificación formal de algoritmos
 - Pre-condiciones, Post-condiciones, Invariantes
 - Eficiencia algorítmica, Notación asintótica
 - Iteración y recursividad
3. Paradigmas Algorítmicos
 - Algoritmos voraces
 - Divide y vence
 - Vuelta atrás
 - Programación dinámica
4. Complejidad Computacional: P, NP, ...

Ejes metodológicos de la asignatura

Atendiendo al horario de la asignatura, cada semana el estudiante asiste a 2 horas de Teoría y a 2 horas de laboratorio (PRALAB).

En las sesiones de Teoría se presentan los temas que se pueden consultar en el apartado de contenidos. Incorporan ejemplos ilustrativos y propuestas de problemas para resolver en las clases de laboratorio.

Las sesiones PRALAB se imparten en el laboratorio y presentan problemas y se analizan las soluciones propuestas. También se pueden presentar las prácticas de la asignatura y se realiza el trabajo de laboratorio correspondiente.

El trabajo autónomo del estudiante consiste en la resolución de los ejercicios propuestos y las tareas de prácticas cuando así se indique.

El lenguaje de programación usado en las prácticas es Python 3.9+.

Plan de desarrollo de la asignatura

Sem	Descripción	Actividad Teoría	Actividad PRALAB	Trabajo autónomo
1	Preliminares	T1: Preliminares	Repaso Python	Consulta de bibliografía y programa, Repaso Python
2	Especificación	T2: Especificación	Problemas T2	Problemas T2
3	Especificación	T2: Especificación	Problemas T2, Presentación P1	P1, Problemas T2
4	Eficiencia algorítmica	T2: Especificación	P1	P1, Problemas T2
5	Iteración	T2: Especificación	P1	P1, Problemas T2
6	Recursividad	T2: Especificación	Problemas T2	P1, Problemas T2
7	Recursividad	T2: Especificación	P1	P1, Problemas T2
8	Paradigmas: Intro	T3: Paradigmas	Entrega P1, Dudas T2	Estudiar
9		1r Parcial		Estudiar
10	Algoritmos voraces	T3: Paradigmas	Presentación P2	P2, Problemas T3
11	Divide y vence	T3: Paradigmas	Problemas T3	P2, Problemas T3
12	Vuelta atrás	T3: Paradigmas	P2	P2, Problemas T3
13	Programación dinámica	T3: Paradigmas	P2	P2, Problemas T3
14	Complejidad	T4: Complejidad	Problemas T3, T4	P2, Problemas T4

Sem	Descripción	Actividad Teoría	Actividad PRALAB	Trabajo autónomo
15	Complejidad	T4: Complejidad	Entrega P2, Dudas T3 y T4	Problemas T3
16/17		2o Parcial		Estudiar
18				
19		Recuperación		Estudiar

Sistema de evaluación

Acr	Actividad de evaluación	Ponderación	Nota Mínima	En grupo	Obligatoria	Recuperable
PE1	Examen 1r Parcial	25%	-	No	No	Sí
PE2	Examen 2o Parcial	25%	-	No	No	Sí
P1	Práctica 1	25%	-	Sí (<=2)	No	No
P2	Práctica 2	25%	-	Sí (<=2)	No	No

$$\text{Nota Final} = 0.25 PE1 + 0.25 PE2 + 0.25 P1 + 0.25 P2$$

Recuperación de las pruebas escritas 1 y 2: Si la nota final obtenida en la asignatura es <5, entonces el estudiante puede optar a mejorar/recuperar el 50% que representen las pruebas escritas (el estudiante podrá elegir qué parte quiere recuperar, o elegir las dos partes).

Salvo nueva situación de excepcionalidad, las pruebas escritas serán presenciales.

Bibliografía y recursos de información

- Gilles Brassard, Paul Bratley. Fundamentals of Algorithmics. Prentice Hall. 1996.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Introduction to Algorithms (3rd ed). MIT Press. 2009.
- Steven S. Skiena. The Algorithm Design Manual. Springer. 2008.
- Cristopher Moore, Stephan Mertens. The Nature of Computation. Oxford University Press. 2011.