

## Análisis y Diseño de Algoritmos

Código: 102783  
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2502441 Ingeniería Informática	OB	3	1
2502441 Ingeniería Informática	OT	4	1

### Contacto

Nombre: Aura Hernandez Sabate

Correo electrónico: [aura.hernandez@uab.cat](mailto:aura.hernandez@uab.cat)

### Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)

Algún grupo íntegramente en inglés: No

Algún grupo íntegramente en catalán: Sí

Algún grupo íntegramente en español: No

### Equipo docente

Javier Sánchez Sánchez Pujadas

### Prerequisitos

No hay prerequisitos formales, pero se recomienda haber aprobado las siguientes asignaturas:

Primer curso del grado:

- Fundamentos de informática
- Metodología de la programación
- Matemática discreta

Segundo curso del grado:

- Laboratorio de programación

### Objetivos y contextualización

Esta asignatura es la continuación de las asignaturas de programación vistas a primero y segundo, como Fundamentos de informática, Metodología de la programación y Laboratorio de programación. Partiendo de la base de que el/la estudiante ya tiene unos conocimientos básicos sobre programación, esta asignatura está centrada en introducir diferentes estilos y paradigmas de diseño de algoritmos. El objetivo principal es que los estudiantes desarrollen habilidades en el diseño y análisis de algoritmos para poder resolver problemas del mundo real de manera efectiva y eficiente de acuerdo con los requisitos establecidos por un cliente potencial.

Por ello se espera que al finalizar el curso el alumnado sabrá:

- Especificar formalmente problemas y programas, y verificarlos.
- Utilizar pruebas formales para validar programas e invariantes para diseñar basándose en contratos.
- Calcular la complejidad algorítmica y computacional de un algoritmo.

Por otra parte conocerá y sabrá elegir en cada momento diferentes estilos y paradigmas de diseño de algoritmos como:

- recursividad
- backtracking
- programación dinámica
- algoritmos probabilísticos
- Etc.

## **Competencias**

- Ingeniería Informática
- Adquirir hábitos de pensamiento.
- 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.
- Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
- Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
- Trabajar en equipo.

## **Resultados de aprendizaje**

1. Conocer los mecanismos de funcionamiento de los diferentes paradigmas de programación.
2. Desarrollar el pensamiento científico.
3. Desarrollar la capacidad de análisis, síntesis y prospectiva.
4. Evaluar la complejidad de los algoritmos y identificar sus puntos débiles.
5. Identificar y seleccionar estrategias algorítmicas adecuadas al problema.
6. Seleccionar la mejor técnica de programación para la resolución de problemas complejos.
7. Trabajar cooperativamente.

## **Contenido**

- Tema 1. Pre-condiciones, post-condiciones e invariantes
- Tema 2. Recursividad y Complejidad Computacional
- Tema 3. Backtracking
- Tema 4. Branch & Bound
- Tema 5. Programación Dinámica
- Tema 6. Greedy
- Tema 7. Algoritmos Probabilísticos
- Tema 8. Análisis de algoritmos

## **Metodología**

Teniendo en cuenta que el objetivo final de la asignatura es que los alumnos desarrollen habilidades para analizar y diseñar algoritmos de forma eficiente según un problema dado, el trabajo del alumnado es el eje central de su aprendizaje, acompañado y guiado por el profesorado. Por este motivo, las sesiones dirigidas serán altamente prácticas y se centrarán en que el alumnado consolide los conocimientos que son objetivo de aprendizaje de esta asignatura.

La metodología general de la asignatura se puede dividir en tres fases:

Preparación de la clase: el objetivo de esta fase es que el alumnado pueda aprender los conceptos que se trabajarán en la sesión siguiente mediante diversas actividades ofrecidas por el profesorado como puede ser el visionado de vídeos, la lectura de textos, etc.

Clase dirigida: el objetivo de esta fase es la de consolidar los conceptos vistos y ponerlos en valor dentro del contexto de la asignatura. El profesorado velará para que el alumnado profundice en estos conceptos mediante ejercicios (más o menos) guiados durante la sesión.

Trabajo autónomo: para que el alumnado tome soltura en la programación de los algoritmos vistos este tendrá que hacer una parte del trabajo por su cuenta con

1. ejercicios sueltos que se entregarán para ser evaluados
2. dentro de un proyecto que se irá realizando a lo largo de todo el curso.

Proyecto de programación: Dentro del trabajo autónomo pedido al alumnado, habrá que realizar un proyecto de programación que se irá desarrollando a lo largo de todo el curso, a medida que se vaya avanzando en el temario. Cada parte del proyecto estará relacionada con uno de los temas previstos y se irán planteando dentro de las sesiones de clase. Se dedicarán algunas horas de las sesiones a plantear el trabajo a realizar y también a hacer el seguimiento del correcto desarrollo, así como el planteamiento de dudas.

**COMPETENCIAS TRANSVERSALES:** En esta asignatura se trabajarán y evaluarán las siguientes competencias transversales:

T01.02 - Desarrollar la capacidad de análisis, síntesis y prospectiva

T01.03 - Desarrollar el pensamiento científico

T03.01 - Trabajar cooperativamente

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

## Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Clases presenciales	50	2	4, 1, 3, 5, 6, 7
Tipo: Supervisadas			
Refuerzo y seguimiento en la resolución del proyecto y los ejercicios	4	0,16	4, 3, 5, 6
Seguimiento en la asimilación de los conceptos teóricos	4	0,16	4, 1, 3, 5, 6
Tipo: Autónomas			
Desarrollo de un proyecto	36	1,44	4, 1, 2, 3, 5, 7
Elaboración de informes prácticos	8	0,32	2, 3, 7
Preparación de parciales	12	0,48	4, 1, 3, 5, 6
Preparación previa a las clases	12	0,48	2, 3, 5, 6
Trabajo autónomo	12	0,48	4, 1, 2, 3, 5, 6, 7

## Evaluación

Se evaluarán tres tipos de actividades de manera independiente y la suma ponderada de ellas dará la nota final. Estas tres actividades son:

- Pruebas de síntesis (PS)
- Ejercicios evaluables (EA)
- Proyecto práctico (P)

1. La primera parte (PS) consiste en la realización de dos pruebas parciales en las que se evaluará al alumnado de forma individual. La nota mínima para aprobar cada prueba es de 4, pero la media de las dos deberá llegar al 5.

2. La segunda parte (EA) consistirá en la entrega de actividades a lo largo de todo el curso. Estas actividades pueden ser en forma de ejercicios, cuestionarios, informes, etc y se pueden plantear dentro de las sesiones de clase o pueden ser propuestas como deberes para casa. La nota final saldrá de la suma ponderada de las entregas fijadas que se pidan.

3. La tercera parte (P) se evaluará de manera grupal (con la entrega de un proyecto) e individual (con la evaluación de una prueba escrita). La nota final se obtendrá de la suma ponderada de la nota grupal y la individual. El proyecto consta de diferentes entregas a lo largo del curso las notas de las cuales conformarán la nota grupal. La nota mínima para aprobar el proyecto es de 5, mientras que el examen individual hay que aprobarlo con una nota mínima de 3.5. La nota final de esta parte deberá ser como mínimo un 5.

Para aprobar la asignatura es necesario que la evaluación de cada una de las partes supere el mínimo exigido y que la evaluación total supere los 5 puntos.

### NOTA DEL EXPEDIENTE, MATRÍCULA DE HONOR Y NO EVALUABLE

En caso de no superar la asignatura debido a que alguna de las actividades de evaluación no alcanza la nota mínima requerida, la nota numérica del expediente será la nota máxima entre los parciales suspendidos, o en su defecto, la nota suspendida máxima. Con las excepciones de los estudiantes que:

- 1) no participen en ninguna de las actividades de evaluación, que se otorgará la calificación de "no evaluable" (cualquier alumno que entregue una práctica o una evaluación programada tendrá nota),
- 2) hayan cometido irregularidades en un acto de evaluación, que se otorgará el valor menor entre 3.0 y la nota numérica máxima antes citada (y por tanto no será posible el aprobado por compensación).

Se darán tantas matrículas como puedan dentro de la normativa de la universidad, siempre y cuando la nota mínima sea un 9.

### RECUPERACIÓN

PS: En el caso de suspender o no presentarse a alguna de las pruebas individuales se podrán recuperar el día asignado a la semana oficial de exámenes.

EA: Esta parte no tendrá la posibilidad de recuperarse.

P: En el caso de suspender la evaluación individual del proyecto (es necesario haberse presentado la primera vez), se podrá recuperar el día asignado en la semana oficial de exámenes. Las entregas parciales del proyecto se pueden recuperar en las siguientes entregas, con una nota final del 80% (y por lo tanto nunca superior a 8).

### CONVALIDACIÓN

No se convalidan proyectos de años anteriores.

PLAGIOS, COPIAS, ETC:

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un estudiante que puedan conducir a una variación de la calificación se calificarán con un cero (0). Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera de estas actividades de evaluación para aprobar la asignatura, esta asignatura quedará suspendida directamente, sin oportunidad de recuperarla en el mismo curso. Estas irregularidades incluyen, entre otros:

- la copia total o parcial de una práctica, informe, o cualquier otra actividad de evaluación;
- dejar copiar;
- presentar un trabajo de grupo no hecho íntegramente por los miembros del grupo;
- presentar como propios materiales elaborados por un tercero, aunque sean traducciones o adaptaciones, y en general trabajos con elementos no originales y exclusivos del estudiante;
- tener dispositivos de comunicación (como teléfonos móviles, smart watches, etc.) accesibles durante las pruebas de evaluación teórico-prácticas individuales (exámenes).

En resumen: copiar, dejar copiar o plagiar en cualquiera de las actividades de evaluación equivale a un SUSPENSO con nota inferior a 3.5.

## COMUNICACIÓN

Las fechas de evaluaciones y entregas se publicarán en el gestor documental Caronte (<https://caronte.uab.cat/>) y pueden estar sujetos a posibles cambios de programación por motivos de adaptación a posibles incidencias. Siempre se informará al Caronte sobre estos cambios ya que se entiende que esta es la plataforma habitual de intercambio de información entre profesores y estudiantes.

## Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Evaluación grupal del proyecto	30	2	0,08	4, 1, 2, 3, 5, 6, 7
Evaluación de actividades desarrolladas en el aula	10	1	0,04	4, 1, 2, 3, 5, 6, 7
Evaluación de actividades desarrolladas fuera del aula	20	1	0,04	4, 1, 2, 3, 5, 6, 7
Evaluación individual del proyecto	10	1	0,04	2, 3
Examen Final (recuperación)	30	3	0,12	4, 1, 2, 3, 5, 6
Primer Examen Parcial Teórico-Práctico	15	2	0,08	4, 1, 2, 3, 5
Segundo Examen Parcial Teórico-Práctico	15	2	0,08	4, 1, 2, 3, 5, 6

## Bibliografía

- Brassard, G., Bratley, P., & Garcia-Bermejo, R. (1997). *Fundamentos de algoritmia* (Vol. 86). Madrid: Prentice Hall.
- Guerequeta, R., & Vallecillo, A. (2019). *Técnicas de diseño de algoritmos*.
- Villegas Jaramillo, & Guerrero Mendieta, L. E. (2016). *Análisis y diseño de algoritmos: un enfoque práctico* / Eduardo Villegas Jaramillo, Luz Enith Guerrero Mendieta. (Primera edición.). Disponible en línea
- Eckel, B. (2021). *Thinking in C++*.
- Stroustrup, B. (1993). *C++ el lenguaje de Programación*. Eddison Wesley/Díaz de Santos.
- <http://en.cppreference.com/w/>
- <http://www.cplusplus.com/>

## Software

Se programará en C ++ básico y se utilizará Microsoft Visual Studio 2019 (licencia UAB). Para la instalación de VS2019 se utilizarán las siguientes opciones:

- Microsoft Visual Studio Community 2019
- Desktop development with C ++
- C ++ MFC latest v142