



Universidad
de Alcalá

GUÍA DOCENTE

Algoritmia y Complejidad

Grado en
Ingeniería Informática (GII)

Universidad de Alcalá

Curso Académico 2022/2023

2º Curso - 2º Cuatrimestre (GII)

GUÍA DOCENTE

Nombre de la asignatura:	Algoritmia y Complejidad
Código:	780021 (GII)
Titulación en la que se imparte:	Grado en Ingeniería Informática (GII)
Departamento y Área de Conocimiento:	Ciencias de la Computación Ciencias de la Computación
Carácter:	Obligatoria (GII)
Créditos ECTS:	6.0
Curso y cuatrimestre:	2º Curso - 2º Cuatrimestre (GII)
Profesorado:	Por definir
Horario de Tutoría:	Consultar al comienzo de la asignatura
Idioma en el que se imparte:	Español

1a. PRESENTACIÓN

La algoritmia es uno de los pilares de la programación y su relevancia se muestra en el desarrollo de cualquier aplicación, más allá de la mera construcción de programas. Esta asignatura trata sobre el análisis y diseño de algoritmos, y pretende exponer al alumno las técnicas necesarias para su diseño e implementación, así como presentar las herramientas que le permitan medir su efectividad y eficiencia.

El objetivo de la asignatura no es el de aportar al alumno soluciones conocidas a problemas concretos, si no proveer al alumno de las técnicas algorítmicas básicas que le permitirán abordar el desarrollo de programas correctos y eficientes para resolver problemas no triviales, estudiando en el desarrollo de la asignatura las técnicas y patrones de diseño de algoritmos más importantes (familias de algoritmos), y profundizando en el diseño y evaluación de los algoritmos propiamente dichos.

Prerrequisitos y Recomendaciones

Debido a que la asignatura presenta diferentes técnicas y estructuras de datos avanzadas para la resolución de problemas, el alumno debe conocer y tener práctica en programación imperativa, y especialmente en el uso de la recursión. Por tanto, es recomendable que el alumno haya superado con éxito las asignaturas de Fundamentos de Programación y Estructuras de Datos.

En cuanto a la parte matemática de la asignatura, el alumno deberá estar familiarizado con los conceptos de límite, combinatoria y resolución de sistemas de ecuaciones, por lo que las asignaturas de Fundamentos Matemáticos y Matemática Discreta deberían haberse cursado con éxito

1b. COURSE SUMMARY

Complexity and Algorithms is a compulsory subject that is taught in the first semester of the third year of the Degree in Informatics Engineering.

The subject introduces the students in the fundamental principles and technics about the design of algorithms and the analysis of their complexity i.e. about the methods for solving problems by means of computers and the cost of their implementation, including the study of the effectiveness and efficiency of the methods.

For a good profit, the student must have good knowledge and skills on the subjects of Programming and Data structure, as much as mathematical reasoning and Discrete Mathematics.

2. COMPETENCIAS

Competencias básicas, generales y transversales.

Esta asignatura contribuye a adquirir las siguientes competencias básicas, generales y transversales:

CB1 - Que los estudiantes hayan demostrado poseer y comprender conocimientos en un 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.

CB2 - Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB3 - Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes

(normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

CB4 - Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5 - Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

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.

TRU1 - Capacidad de análisis y síntesis.

TRU2 - Comunicación oral y escrita.

TRU3 - Capacidad de gestión de la información.

TRU4 - Capacidad de aprendizaje autónomo.

TRU5 - Capacidad para trabajar en equipo.

Competencias Específicas

Esta asignatura proporciona la(s) siguiente(s) competencia(s) específica(s):

CC1 - Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.

CC3 - 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.

CC4 - Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.

Resultados de aprendizaje

Al terminar con éxito esta asignatura/enseñanza, los estudiantes serán capaces de:

RA1. Razonar los fundamentos teóricos de los esquemas algorítmicos.

RA2. Aprender a realizar el análisis de un algoritmo, atendiendo a su eficiencia, costes, casos, notación asintótica, etc., entendiéndolo como un criterio de calidad.

RA3. Aplicar con criterio las técnicas básicas de análisis de eficiencia y diseño de algoritmos.

RA4. Analizar la corrección de un algoritmo usando técnicas sencillas de verificación.

RA5. Reconocer los esquemas algorítmicos básicos: Divide y vencerás, voraces, programación dinámica, algoritmos de vuelta atrás, ramificación y acotación.

RA6. Conocer algunos algoritmos clásicos para problemas fundamentales, y reconocer las situaciones donde poder reutilizar el repertorio de estructuras de datos y algoritmos clásicos.

RA7. Escribir programas recursivos sobre tipos no necesariamente básicos y poder razonar sobre su corrección y eficiencia.

RA8. Ser capaz de utilizar técnicas avanzadas de diseño y análisis de algoritmos, y saber particularizar esquemas algorítmicos generales para resolver problemas.

RA9. Disponer de criterios que permitan, durante las etapas de especificación, diseño e implementación escoger la alternativa más adecuada, y disponer de elementos por argumentar de forma razonada las elecciones realizadas.

RA10. Tomar contacto con algunas técnicas fundamentales de diseño y análisis de algoritmos, así como con algunas técnicas avanzadas de programación.

RA11. Saber identificar las componentes más relevantes de un problema y seleccionar la técnica algorítmica más adecuada para su resolución.

RA12. Ser capaz de seleccionar los tipos de datos más adecuados para mejorar la eficiencia de una solución algorítmica.

RA13. Diseñar justificadamente soluciones algorítmicas eficientes de problemas propios del nivel académico de la asignatura.

3. CONTENIDOS

Introducción a la algoritmia Introducción a la algoritmia	12 horas
Algoritmos básicos Algoritmos voraces Algoritmos divide y vencerás Algoritmos de programación dinámica	24 horas
Algoritmos avanzados Uso de recursión intensiva	14 horas
No determinismo y Clases de complejidad Algoritmos no deterministas Complejidad	6 horas

4. METODOLOGÍAS DE ENSEÑANZA APRENDIZAJE. ACTIVIDADES FORMATIVAS

4.1. Distribución de créditos

Número de horas presenciales:	58 horas (56 horas de clase presencial +2 horas de evaluación)
Número de horas del trabajo propio del estudiante:	92 (Incluye horas de estudio, elaboración de actividades, preparación de exámenes)
Total horas	150

4.2. Estrategias metodológicas, materiales y recursos didácticos

Clases magistrales y expositivas, en combinación con prácticas en el laboratorio	Exposición y discusión de los conocimientos básicos de la asignatura. Planteamiento y resolución teórica de ejercicios y supuestos relacionados. Orientadas a la enseñanza de las competencias específicas de la asignatura, especialmente las relacionadas con las técnicas de creación y mejora de algoritmos.
Trabajos en grupo y cooperativo	Planteamiento y desarrollo de ejercicios prácticos que contribuyan al entendimiento de la materia y al desarrollo de la práctica en el análisis de problemas, razonamiento crítico y comprensión de los métodos de resolución.
Trabajo y estudio personal	Análisis y asimilación de los contenidos de la materia, resolución de problemas, consulta bibliográfica, preparación de trabajos individuales y grupales, realización de exámenes presenciales y autoevaluaciones. Orientadas especialmente al desarrollo de métodos para la autoorganización y planificación del trabajo individual y en equipo.

5. EVALUACIÓN: Procedimientos, criterios de evaluación y calificación

Preferentemente se ofrecerá a los alumnos un sistema de evaluación continua que tenga características de evaluación formativa de manera que sirva de realimentación en el proceso de enseñanza-aprendizaje por parte del alumno.

5.1. PROCEDIMIENTOS

La evaluación debe estar inspirada en los criterios de evaluación continua (Normativa de Evaluación de los Aprendizajes, NEA, art 3). No obstante, respetando la normativa de la Universidad de Alcalá se pone a disposición del alumno un proceso alternativo de evaluación final de acuerdo a la [Normativa de Evaluación de los Aprendizajes](#) según lo indicado en su Artículo 10, los alumnos tendrán un plazo de quince días desde el inicio del curso para solicitar por escrito al Director de la Escuela Politécnica Superior su intención de acogerse al modelo de evaluación no continua aduciendo las razones que estimen convenientes. La evaluación del proceso de aprendizaje de todos los alumnos que no cursen solicitud al respecto o vean denegada la misma se realizará, por defecto, de acuerdo al modelo de evaluación continua. El estudiante dispone de dos convocatorias para superar la asignatura, una ordinaria y otra extraordinaria.

Convocatoria ordinaria

Evaluación continua:

Las principales herramientas de evaluación serán:

- 1.-**Prueba de evaluación intermedia (PEI 1)** De carácter escrito, consistente en la resolución de problemas y cuestiones sobre los bloques temáticos 1 y 2.
- 2.-**Prueba de evaluación intermedia (PEI 2)** De carácter escrito, consistente en la resolución de problemas y cuestiones sobre los bloques temáticos 3 y 4
- 3.-**Prueba de evaluación intermedia de laboratorio (PEIL 1)**, consistente en la entrega y defensa de una serie de ejercicios relativos a los conocimientos impartidos en clase de laboratorio, como aplicación práctica de los mismos.

Evaluación mediante examen final:

En el caso de evaluación mediante examen final, los elementos de evaluación a emplear serán los siguientes:

1. **Prueba de evaluación final (PEF)** De carácter escrito, consistente en la resolución de problemas y cuestiones sobre los cuatro bloques temáticos de la asignatura y de sus prácticas de laboratorio.

Convocatoria extraordinaria

El procedimiento será el mismo que el descrito para la evaluación mediante examen final en la convocatoria ordinaria.

5.2. EVALUACIÓN

CRITERIOS DE EVALUACIÓN

Se utilizarán los siguientes criterios para la evaluación de la asignatura, relacionados con los resultados del aprendizaje:

- CE1.** Comprensión y uso de los conceptos y técnicas básicas sobre algoritmos, diseño, análisis, eficiencia, y coste de los algoritmos
- CE2.** Comprensión y uso de los elementos, técnicas y uso de los algoritmos básicos sus estructuras de datos, estrategias básicas, voracidad, eficiencias usuales, reutilización de casos y programación dinámica.
- CE3.** Comprensión y uso de técnicas de recursión intensiva, ventajas e inconvenientes, problemas de ramificación y poda.
- CE4.** Comprensión y uso del no determinismo y de métodos probabilísticos, validez de resultados, ventajas e inconvenientes particulares, así como del análisis fundamental de los problemas de complejidad y de la irresolubilidad algorítmica de determinados problemas.
- CE5.** Capacidad de aplicación práctica en casos prácticos programados de laboratorio.

INSTRUMENTOS DE EVALUACIÓN

Esta sección resume los instrumentos de calificación que serán aplicados a cada uno de los criterios de Evaluación.

- **Prueba de evaluación intermedia (PEI 1)** De carácter escrito, consistente en la resolución de

problemas y cuestiones sobre los bloques temáticos 1 y 2.

- **Prueba de evaluación intermedia (PEI 2)** De carácter escrito, consistente en la resolución de problemas y cuestiones sobre los bloques temáticos 3 y 4
- **Prueba de evaluación intermedia de laboratorio (PEIL 1)**, consistente en la entrega y defensa de una serie de ejercicios relativos a los conocimientos impartidos en clase de laboratorio, como aplicación práctica de los mismos.
- **Prueba de evaluación final (PEF)** De carácter escrito, consistente en la resolución de problemas y cuestiones sobre los cuatro bloques temáticos de la asignatura y de sus prácticas de laboratorio.

CRITERIOS DE CALIFICACIÓN

En la convocatoria **ordinaria–evaluación continua** la relación entre las competencias, resultados del aprendizaje, criterios e instrumentos de evaluación, es la siguiente.

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CG8, CG9 CC1, CC3	RA1-RA7, RA11	CE1-CE4	PEI-1	30%
CG8, CG9 CC1, CC3, CC4	RA8-RA13	CE1-CE4	PEI-2	35%
CG8, CG9 CC1, CC3,CC4	RA1-RA13	CE5	PEIL-1	35%

Se otorgará la calificación de "No presentado" al alumno que habiendo optado por el procedimiento de evaluación continua, cumpla alguno de los siguientes requisitos:

- Cuando el alumno no haya entregado, al menos el 60% de los trabajos solicitados.

En la convocatoria **ordinaria–evaluación final** la relación entre las competencias, resultados del aprendizaje, criterios e instrumentos de evaluación, es la siguiente.

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CG1-CG8 CC1, CC3, CC4	RA1-RA13	CE1-CE4	PEF	100%

Convocatoria extraordinaria

En el caso de la convocatoria extraordinaria se mantendrán los mismos porcentajes que se han establecido en el caso de la evaluación mediante examen final, dando la opción de realizar la PEIL1 o de mantener la nota obtenida en la evaluación continua, según decisión del alumno.

6. BIBLIOGRAFÍA

6.1. Bibliografía básica

- A.V. Aho, J.E. Hopcroft, J.D. Ullman, Estructuras de datos y algoritmos, Addison-Wesley Iberoamericana, 1988.
- G.Brassard, P. Bratlet, Fundamentos de algoritmia, Prentice Hall, 1999.
- G.H. Gonnet, R. Baeza-Yates, Handbook of Algorithms and Data Structures, AddisonWesley, 1991.
- M.A. Weiss, Estructuras de datos y algoritmos, Addison-Wesley Iberoamericana, 1995.

6.2. Bibliografía complementaria

- T. Cormen, C. Leirserson, R. Rivest, Introduction to Algorithms, McGraw Hill, 1998.
- R. Peña, Diseño de Programas. Formalismo y Abstracción. Prentice Hall, 1997.
- N. Wirth, Algoritmos + Estructuras de datos = Programas. Editorial del Castillo, 1984.
- N. Ziviani, Diseño de algoritmos con implementaciones en Pascal y C, Thompson, 2007.

NOTA INFORMATIVA

La Universidad de Alcalá garantiza a sus estudiantes que, si por exigencias sanitarias las autoridades competentes impidieran la presencialidad total o parcial de la actividad docente, los planes docentes alcanzarían sus objetivos a través de una metodología de enseñanza-aprendizaje y evaluación en formato online, que retornaría a la modalidad presencial en cuanto cesaran dichos impedimentos.