

## DATOS DE IDENTIFICACIÓN

Titulación:	Grado en Ingeniería Informática		
Rama de Conocimiento:	Ingeniería y Arquitectura		
Facultad/Escuela:	Escuela Politécnica Superior		
Asignatura:	Complejidad Computacional		
Tipo:	Obligatoria	Créditos ECTS:	6
Curso:	3	Código:	5630
Periodo docente:	Quinto semestre		
Materia:	Computación		
Módulo:	Tecnología Específica		
Tipo de enseñanza:	Presencial		
Idioma:	Castellano		
Total de horas de dedicación del alumno:	150		

Equipo Docente	Correo Electrónico
Roberto Rodríguez Galán	roberto.rodriguez@ufv.es
José Ignacio Catalina Benavente	jignacio.catalina@ufv.es

## DESCRIPCIÓN DE LA ASIGNATURA

La asignatura de Complejidad Computacional describe los distintos métodos de análisis y diseño de algoritmos y su aplicación en el desarrollo de soluciones que, en última instancia, deberán poder ser expresadas con un lenguaje de programación. Para ello, la asignatura estudia la complejidad computacional de los distintos métodos algorítmicos, determinando en el proceso cual es el mejor en cada caso.

Esta asignatura corresponde al módulo Tecnología Específica y, dentro de éste, a la materia Computación. Se imparte en el primer semestre del tercer curso de los estudios de Grado en Ingeniería Informática y requiere de una dedicación de 150 horas por parte del alumno.

La asignatura abarca la presentación de los principales paradigmas algorítmicos, que serán aplicados a la resolución de problemas básicos en computación como pueden ser la ordenación de datos, el recorrido de árboles y grafos, el problema de la mochila, así como la generación y la resolución automáticas de laberintos, por citar solo algunos, con un especial interés en el estudio de la complejidad de dichos problemas.

La mayoría de los contenidos de esta asignatura son metodologías para la resolución óptima de problemas que se dan en la vida real, incluso para aquellos que computacionalmente no tienen solución a día de hoy, como el problema del viajante de comercio. Estos retos y planteamientos provocarán la curiosidad del alumno y le animarán a investigar distintas soluciones a problemas existentes en el plano humano y técnico de una sociedad cambiante social y tecnológicamente.

## OBJETIVO

El principal objetivo de la asignatura de Complejidad Computacional es describir las bases teóricas de la complejidad temporal de los algoritmos. A partir de este conocimiento teórico, se estudia la forma de calcular la complejidad para un algoritmo concreto. Como conclusión se estudian distintos métodos de análisis y diseño de algoritmos y su aplicabilidad en el desarrollo de soluciones a problemas genéricos (ordenación, búsqueda, etc.). El objetivo final es que el alumno, a partir de la base teórica recibida y los ejercicios prácticos realizados pueda estudiar la complejidad computacional de sus soluciones algorítmicas permitiéndole elegir la mejor opción entre las posibles en base a distintas estrategias de implementación.

Los fines específicos de la asignatura son:

Conocer y comprender las bases teóricas y los problemas históricos de la Complejidad Computacional, como es el problema "¿P=NP?"

Analizar la complejidad de un problema y diseñar el algoritmo que lo resuelve con menor coste computacional posible.

Aplicar diversas técnicas para la optimización computacional de los algoritmos: Divide y Vencerás mediante algoritmos recursivos, Algoritmos voraces, Programación Dinámica, Vuelta Atrás, etc.

## CONOCIMIENTOS PREVIOS

Se recomienda haber cursado las asignaturas de "Matemática Discreta", "Introducción a la Programación", "Matemáticas para la Ingeniería" y "Programación Orientada a Objetos" de primer curso y "Estructuras de Datos y Algoritmos" e "Ingeniería del Software I" de segundo curso.

## CONTENIDOS

Tema 1. Bases teóricas de la Complejidad Computacional:

- Introducción a la complejidad computacional y a los paradigmas algorítmicos. Problema "¿P=NP?".
- Eficiencia y complejidad. Clases de complejidad computacional. Casos mejor, peor y medio. Cotas de complejidad: Notación O, Omega, Theta.
- Ecuaciones de recurrencias.

Tema 2. Casos de estudio algorítmico: Ordenación, Torres de Hanoi, Factorial, etc.

Tema 3. Paradigmas computacionales: Paradigma Divide y Vencerás. Esquema general del paradigma Divide y Vencerás. Algoritmos estudiados: búsqueda binaria, potencia entera de un número, números de Fibonacci (versión DyV), multiplicación de matrices cuadradas (algoritmo clásico, versión DyV ingenua, algoritmo de Strassen), ordenamiento QuickSort y MergeSort, estadístico de orden  $i$ .

Tema 4. Paradigmas computacionales: Paradigma Voraz. Esquema general del paradigma Voraz. Algoritmos estudiados: devolución de cambio, problema de la mochila (versión voraz), recubrimiento mínimo de un grafo (algoritmos de Prim y de Kruskal), camino mínimo en un grafo (algoritmo de Dijkstra), el viajante de comercio.

Tema 5. Paradigmas computacionales: Paradigma de Programación Dinámica. Esquema general del paradigma de Programación Dinámica. Algoritmos estudiados: números de Fibonacci (versión PD), coeficientes binomiales, problema de la devolución de cambio (versión PD), problema de la mochila (1,0) (versión PD).

Tema 6. Paradigmas computacionales: Paradigma de Vuelta Atrás. Esquema general del paradigma de Vuelta Atrás. Algoritmos estudiados: búsqueda en profundidad, resolución de laberintos, recorrido del rey en el ajedrez, problema de las  $n$  reinas.

LAS TRANSPARENCIAS QUE UTILIZA EL PROFESOR EN CLASE SON LA FORMA DE EXPLICAR LA MATERIA, PERO NO SE CONSIDERA EL CONTENIDO DE LA MATERIA. EL CONTENIDO DEL CURSO SE BASA EN LA BIBLIOGRAFIA BASICA INDICADA SIGUIENDO LAS INDICACIONES DEL PROFESOR PARA CADA TEMA.

EN CUALQUIER CASO, Y A LA DISCREPCIONALIDAD DEL PROFESOR, EN EL AULA VIRTUAL NO SE ENTREGARÁN LAS TRANSPARENCIAS UTILIZADAS EN CLASE HASTA EL FINAL DEL CAPITULO O APARTADO CORRESPONDIENTE (SI EL PROFESOR CONSIDERA QUE DEBE ENTREGARLAS). EL ALUMNO ES RESPONSABLE DE TOMAR LOS APUNTES QUE CONSIDERE Y COMPLETAR LA INFORMACION CON LA BIBLIOGRAFIA INDICADA PARA DISPONER DEL CONTENIDO DEL CURSO.

## ACTIVIDADES FORMATIVAS

La enseñanza-aprendizaje del módulo Matemáticas Avanzadas y Computación atiende a una combinación armónica entre el trabajo presencial y autónomo del alumno.

La asignatura de Complejidad Computacional tiene una base teórica para luego su aplicación práctica. La toma de contacto con cada tema se realizará a través de metodologías expositivas por parte del profesor, que esencialmente están dirigidas a presentar los fundamentos teóricos de los distintos paradigmas algorítmicos, así como del estudio de los principales algoritmos de cada tipo. Las clases expositivas serán complementadas con clases de resolución de ejercicios en el aula y como trabajo personal del alumno.

Para garantizar la comprensión de los algoritmos explicados, se plantearán ejercicios prácticos de realización individual y autónoma fuera del aula, donde los alumnos tendrán que mostrar la secuencia ordenada de pasos que supone la ejecución de cada algoritmo.

Asimismo, se plantearán prácticas por equipos de programación de los algoritmos más representativos, para las que deberán aplicarse los conocimientos teóricos aprendidos para su óptima implementación. Dichas prácticas se comenzarán a realizar en clases de orientación práctica, pero deberán completarse de manera autónoma por parte de los alumnos fuera del aula. Para la resolución de dudas y dificultades compartidas por diversos alumnos, el profesor podrá organizar tutorías individuales o grupales.

Para favorecer la adquisición de habilidades de comunicación oral y escrita, así como del vocabulario específico de la asignatura, los alumnos se organizarán en equipos para realizar un trabajo sobre un tema teórico/práctico de la asignatura, del que entregarán una memoria escrita y que tendrán que defender ante la clase.

Finalmente, con el fin de facilitar al alumno el acceso a los materiales y la planificación de su trabajo, así como la comunicación con el profesor y el resto de los alumnos, se empleará el Aula Virtual, la plataforma de aprendizaje online que ofrece diferentes recursos electrónicos para complementar, de forma muy positiva, el aprendizaje del alumno. Allí se dispondrá de los materiales de la asignatura, se plantearán las tareas que deben entregar los alumnos, se añadirán enlaces de interés sobre la asignatura y se habilitará un foro de comunicación entre el profesor y los alumnos.

Sobre una base teórica de 14 semanas en un cuatrimestre, con 4 horas de clase por semana (teórica o práctica), se planifican las siguientes actividades formativas:

- Primer día de Clase: Presentación Presentación asignatura
- Semanas 1, 2 y 3: Teoría Tema 1: Introducción a la Complejidad. Cotas. Medidas asintóticas. Recurrencia. Algoritmos de Ordenación

- Semanas 4 y 5 Práctica 1 Tema 1: Operaciones Elementales. Algoritmos de Ordenación
- Semana 6 Teoría Tema 2: Divide y Vencerás
- Semana 7 y 8 Práctica 2 Tema 2: Divide y Vencerás
- Semana 9 Teoría Tema 3: Paradigma Voraz
- Semana 10 Práctica 3A Tema 3: Paradigma Voraz
- Semana 11 Teoría Tema 4: Programación Dinámica
- Semana 12 Práctica 3B Tema 4: Programación Dinámica
- Semana 13 Teoría Tema 5: Vuelta Atrás
- Semana 14 Presentaciones.

Además de las actividades formativas anteriores, se planteará un trabajo de asignatura realizado por equipos de alumnos (a definir el tamaño del equipo) sobre un libro de distintas temáticas (a elegir por cada grupo) que tenga que ver con el contenido de la asignatura. Este trabajo constará de una lectura profunda del libro elegido, una memoria escrita sobre el mismo y una defensa de esa memoria en clase en la que tendrán que intervenir todos los miembros del equipo.

## DISTRIBUCIÓN DE LOS TIEMPOS DE TRABAJO

ACTIVIDAD PRESENCIAL	TRABAJO AUTÓNOMO/ACTIVIDAD NO PRESENCIAL
68 horas	82 horas
Lecciones expositivas 28h Clases prácticas y laboratorio 28h Presentación de trabajos 4h Tutorías 4h Exámenes 4h	Estudio y trabajo individual 46h Trabajo en grupo 36h

## COMPETENCIAS

### Competencias básicas

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

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

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

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

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

## Competencias generales

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 lo establecido en el apartado 5 de esta memoria.

Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de esta memoria.

## Competencias específicas

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.

## RESULTADOS DE APRENDIZAJE

Ser capaz de analizar la complejidad algorítmica de problemas reales que se dan en el ámbito de la ingeniería.

Ser capaz de analizar la complejidad de los algoritmos (caso mejor, promedio y peor)

Conocer los conceptos subyacentes a los Algoritmos probabilísticos analizando en qué situaciones de la vida real es preciso acudir a este tipo de procedimientos

Ser capaz de aplicar las técnicas de programación dinámica, vuelta atrás y de ramificación y poda.

## SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

### SISTEMA ORDINARIO DE EVALUACIÓN.

Para la evaluación de la asignatura se tendrán en cuenta diversas pruebas con diferentes pesos:

[1] Una **Prueba escrita** de carácter teórico-práctico, y que se realizará al final del curso: **40%** de la calificación final. Será una prueba orientada a evaluar los conocimientos adquiridos por el alumno sobre la complejidad computacional de algoritmos y los distintos paradigmas. El examen se puntuará de 0 a 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario. Se evaluará la expresión escrita, el uso correcto de vocabulario específico de la asignatura, planteamiento de los problemas, corrección, completitud, presentación e interpretación de los resultados obtenidos y la justificación de las respuestas e ideas expresadas con las palabras propias del alumno. Los exámenes se realizarán sin transparencias, apuntes, libros ni cualquier otro material relacionado con la asignatura. El examen se realizará de manera presencial. Para aprobar es necesario obtener un 5.

[2] **Examen práctico** de programación de diversos algoritmos y realizadas de manera autónoma dentro y fuera del horario de clase y repartidas a lo largo del curso: **30%** de la calificación final. Se realizarán de forma individual o en grupos (según indicación del profesor) y consistirán en la implementación en algún lenguaje de programación de diversos algoritmos típicos de cada paradigma. Se evaluará la memoria explicativa (presentación, expresión escrita, uso correcto del vocabulario específico de la asignatura, etc.), así como la calidad, corrección y completitud

del código fuente entregado, incluyendo el uso de abundantes comentarios del código y el uso de estándares de programación. La puntuación de las prácticas será entre 0 y 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario. Para aprobar es necesario obtener un 4 como mínimo en cada práctica y un 5 de media entre todas. Para la entrega de las practicas habrá un plazo obligatorio fijado en el Aula Virtual. La no entrega en el plazo fijado implica un No Presentado en la práctica. Para superar las prácticas los alumnos deben tener un 80% de asistencia a las clases prácticas; en caso de no superar este porcentaje deberán examinar de una parte práctica de la asignatura para asegurar su participación. En cualquier caso, todo alumno o grupo puede ser llamado por los profesores a una defensa personal o grupal de la práctica entregada.

**ES IMPRESCINDIBLE LA ENTREGA DE TODAS LAS PRACTICAS PARA PODER APROBAR ESTE APARTADO. LA NO ENTREGA DE UNA PRACTICA HACE QUE EL ALUMNO SUSPENDA LA ASIGNATURA Y DEBA PRESENTARLAS, SIGUIENDO LAS INSTRUCCIONES DE LA CONVOCATORIA EXTRAORDINARIA, LA PRACTICA PENDIENTE QUE SERÁ NUEVA Y DEFINIDA EXPRESAMENTE PARA ESTA CONVOCATORIA EXTRAORDINARIA.**

[3] **Defensa oral-escrita** por equipos sobre un tema teórico relacionado con la asignatura: **25%** de la calificación final. Este trabajo constará de una memoria escrita y una defensa ante la clase, en la que tendrán que intervenir todos los miembros. Se evaluará la calidad del trabajo (adecuación al tema, presentación, estructura, uso correcto de la plantilla, expresión escrita y corrección gramatical, uso correcto del vocabulario específico de la asignatura, biografía citada, etc.), así como la de la defensa (presentación, respuestas al profesor, expresión oral y corrección gramatical, etc.). La nota de la memoria será el 75% del total y será común para ambos estudiantes; la nota de la defensa será el otro 25% y será individual, en función de cómo realiza cada uno la presentación. Para la entrega de este trabajo habrá un plazo obligatorio fijado en el Aula Virtual. La no entrega en el plazo fijado implica un No Presentado en el trabajo anual de la asignatura.

**ES IMPRESCINDIBLE LA ENTREGA DE ESTE TRABAJO PARA PODER APROBAR ESTE APARTADO. LA NO ENTREGA DEL TRABAJO HACE QUE EL ALUMNO SUSPENDA LA ASIGNATURA Y DEBA PRESENTARLO, SIGUIENDO LAS INSTRUCCIONES DE LA CONVOCATORIA EXTRAORDINARIA.**

[4] **Participación activa** en la asignatura y realización de ejercicios: **5%** de la calificación final. Esta nota tiene dos componentes: la participación y los ejercicios. Respecto de la participación activa, se evaluará el interés mostrado por el alumno en la asignatura. Concretamente se computará el índice de asistencia a tutorías y revisiones de exámenes, individuales o grupales, el grado de participación activa en las clases, la puntualidad y la actitud en clase, así como el respeto al profesor y a los compañeros. Por otro lado, se tendrá en cuenta la realización de ejercicios sobre la asignatura de manera individual, tanto en el horario de clase como fuera del aula. Los alumnos partirán de una especificación del algoritmo en pseudocódigo y deberán implementarlos paso a paso, sin escatimar detalles ni saltarse pasos, para demostrar que han comprendido el algoritmo y que son capaces de aplicarlo a un problema dado. Para contabilizar esta nota, será imprescindible un porcentaje mínimo de asistencia a clase del 80 %. Para porcentajes inferiores a este valor, la nota será de 0.

**CÁLCULO DE LA NOTA FINAL:** Teniendo en cuenta estas cuatro componentes, la nota final del alumno será un valor entre 0 y 10 y se calculará como:

$0,40 * [1] + 0,30 * [2] + 0,25 * [3] + 0,05 * [4]$ .

#### **RECUPERACIÓN EN CONVOCATORIA EXTRAORDINARIA.**

Las notas de las partes aprobadas a lo largo del curso se guardan. Los alumnos que no hayan alcanzado la nota mínima requerida en alguno de los apartados anteriores podrán optar a una recuperación extraordinaria de las partes suspendidas de la siguiente manera:

[1] Prueba escrita teórico-práctico de recuperación (40% de la nota final). En este caso, la nota obtenida en este examen debe ser 5 como mínimo. En caso de tener que recuperar se realizará un examen que englobe todo el temario del curso. Este examen se aprueba con un 5 como mínimo.

[2] Examen práctico no entregados y/o suspendidos (junto con las prácticas aprobadas, su media equivale al 30% de la nota final). Las practicas planteadas para la convocatoria extraordinaria serán nuevas.

[3] Defensa oral-escrita (25% de la nota final). En este caso la defensa de este, que sigue siendo obligatoria, será en horario de tutorías, o bien en el horario que indique el profesor. Tanto la memoria escrita como la defensa siguen haciéndose por equipo salvo caso excepcional en el que se haga se forma individual a criterio del profesor. El trabajo suspenso se recupera cuando la nota obtenida es inferior a 5 para la memoria escrita y/o inferior a 5 para la defensa.

Como norma general, la nota numérica de los exámenes, prácticas, trabajos y ejercicios se redondeará a una cifra decimal.

#### **ALUMNOS CON DISPENSA ACADÉMICA O EN SEGUNDA MATRÍCULA O SUCESIVAS**

Aquellos alumnos que estén exentos de la obligación de asistir a clase, bien por segunda matrícula en la asignatura o sucesivas, bien por contar con autorización expresa de la Dirección del Grado, tendrán la obligación de realizar exámenes, prácticas, trabajos y ejercicios en los mismos plazos que el resto de sus compañeros. Respecto del porcentaje del 5% correspondiente a participación en la asignatura y realización de ejercicios, serán evaluados mediante la asistencia a un mínimo de una tutoría, en horario convenido entre profesor y alumno. En dicha tutoría el alumno hará entrega de los ejercicios del curso y responderá a las preguntas que le efectúe el profesor sobre ellos, recibiendo una nota que será el 10% restante de la nota final. Habrá obligatoriamente otra tutoría en la que el alumno realizará la defensa del trabajo.

### **NORMATIVA ANTIPLAGIOS.**

Cualquier tipo de fraude o plagio por parte del alumno en una actividad evaluable, será sancionado según se recoge en la Normativa de Convivencia de la UFV. A estos efectos, se considerará "plagio" cualquier intento de defraudar el sistema de evaluación, como copia en ejercicios, exámenes, prácticas, trabajos o cualquier otro tipo de entrega, bien de otro compañero, bien de materiales o dispositivos no autorizados, con el fin de hacer creer al profesor que son propios.

### **NO PRESENTADO.**

A efecto de cómputo de convocatorias en una asignatura, solamente se contabilizarán como consumidas aquellas en las que el alumno se haya presentado a una parte de estas, aunque no se presente al examen final. Se entenderá que un alumno se ha presentado a una prueba, aunque la abandone una vez comenzada la misma. La condición de NO PRESENTADO en la convocatoria extraordinaria estará ligada a la no asistencia o entrega de ninguna prueba, práctica o trabajo que esté pendiente.

## **BIBLIOGRAFÍA Y OTROS RECURSOS**

### **Básica**

Sanjeev Arora and Boaz Barak. Computational Complexity: A Modern Approach. 2007  
Disponible online en <http://theory.cs.princeton.edu/complexity/book.pdf>

S. S. Skiena The Algorithm Design Manual 2.<sup>a</sup> edición  
Londres: Springer Verlag, 2008. ISBN: 978-1-84800-069-8

R. Guerequeta y A. Vallecillo Técnicas de diseño de algoritmos 2.<sup>a</sup> edición  
Servicio de Publicaciones de la Universidad de Málaga, 2000. ISBN: 84-7496-666-3. Disponible online desde la web de los autores: <http://www.lcc.uma.es/~av/Libro/>

### **Complementaria**

N. Martí Oliet, Y. Ortega Mallén y A. Verdejo, Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos, 2.<sup>a</sup> edición  
Garceta Grupo Editorial, 2013. ISBN: 978-84-1545-265-2.

L. Araujo Serna, R. Martínez Unanue y M. Rodríguez Artacho, Programación y estructuras de datos avanzadas 1.<sup>a</sup> edición.  
Editorial Universitaria Ramón Areces, 2011. ISBN: 9788499610221