



## 17827 - ANÁLISIS DE ALGORITMOS

### Información de la asignatura

**Código - Nombre:** 17827 - ANÁLISIS DE ALGORITMOS

**Titulación:** 473 - Graduado/a en Ingeniería Informática  
474 - Graduado/a en Ingeniería Informática y Matemáticas  
722 - Graduado/a en Ingeniería Informática (Modalidad Bilingüe 2018)  
734 - Graduado/a en Ingeniería Informática y Matemáticas (2019)

**Centro:** 350 - Escuela Politécnica Superior

**Curso Académico:** 2020/21

### 1. Detalles de la asignatura

#### 1.1. Materia

Programación y estructuras de datos

#### 1.2. Carácter

Obligatoria

#### 1.3. Nivel

Grado (MECES 2)

#### 1.4. Curso

2

#### 1.5. Semestre

Primer semestre

#### 1.6. Número de créditos ECTS

6.0

#### 1.7. Idioma

Español

#### 1.8. Requisitos previos

Ninguno/None.

#### 1.9. Recomendaciones

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	1/9	

Para un buen aprovechamiento del curso, es recomendable haber aprobado las materias Programación I y Programación II. De no ser así, el curso puede seguirse pero requiriendo tal vez un esfuerzo extra, sobre todo en su parte práctica, lo que puede incidir en el rendimiento del estudiante en otras asignaturas en las que esté matriculado.

También se supone unos conocimientos matemáticos básicos al nivel de los cursos Álgebra y Cálculo I.

En relación a las clases prácticas, se recomienda que el estudiante sea capaz de expresar en pseudocódigo algoritmos básicos que incluyan bucles anidados así como las estructuras de control if, while y for. A su vez, sus conocimientos del lenguaje C deben incluir

- Conocer la estructura general de un programa en C.
- Comprender los tipos de datos primitivos de C y utilizar los tipos estructurados del lenguaje (tablas y estructuras)
- Utilizar correctamente las estructuras de control
- Escribir prototipos y cuerpos de funciones y conocer los mecanismos de llamada por valor y por dirección.
- Tener una cierta familiaridad con los algoritmos recursivos.
- Conocer y aplicar las estructuras de control y selección de flujo en C.
- Conocer y hacer uso de punteros y de la gestión dinámica de memoria.
- Estar familiarizado con el uso de algún entorno gráficos de programación, como Visual C++, Eclipse, NetBeans o similares.

Asimismo, es recomendable que el estudiante disponga de un dominio del idioma inglés que le permita leer la bibliografía de consulta.

#### 1.10. Requisitos mínimos de asistencia

Se considerarán dos itinerarios, Evaluación Continua y No Evaluación Continua, tanto para teoría como para prácticas.

El itinerario de Evaluación Continua en Teoría requerirá una asistencia continuada a las clases teóricas **así como presentarse a los dos parciales que se convoquen (salvo por razones de fuerza mayor); el no cumplir este último requisito supondrá el abandono del itinerario.**

El itinerario de Evaluación Continua en Prácticas requerirá una asistencia mínima del 85% de clases prácticas así como la entrega puntual de las prácticas asignadas.

En cualquier caso, **la asistencia a clases de teoría o problemas se considera esencial para la superación de la asignatura**, ya que dicha asistencia supone

- La toma de contacto explicada con el material de la asignatura, con la que el estudiante obtiene una primera y en general suficiente comprensión de dicho material. Dicha comprensión es **muy difícil y costosa de obtener por otras vías.**
- Una parte muy importante, **casi un 30%, del tiempo que el estudiante debe dedicar a la asignatura** (unas 45 horas sobre 150).

Por todo ello es sumamente recomendable asistir a dichas clases, indicando la experiencia que **no hacerlo convierte en extremadamente difícil la superación de la asignatura.**

#### 1.11. Coordinador/a de la asignatura

Jose Ramon Dorronsoro Ibero

<https://autoservicio.uam.es/paginas-blancas/>

#### 1.12. Competencias y resultados del aprendizaje

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	2/9	

### 1.12.1. Competencias

**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.

### 1.12.2. Resultados de aprendizaje

Dentro de los **resultados del aprendizaje**, el estudiante adquiere en la asignatura Análisis de Algoritmos con carácter general los siguientes:

- Comprensión de los principales algoritmos así como de las técnicas necesarias para la estimación de su complejidad.
- Comprensión de las técnicas de programación utilizando lenguajes de alto nivel.

De manera más detallada, los mismos pueden desarrollarse como sigue:

- Eficacia de algoritmos: Concepto, elementos de medida, casos de medida. Análisis de bucles y selecciones. Análisis general de algoritmos recursivos.
- Herramientas matemáticas básicas del análisis de eficacia de algoritmos: Comparación asintótica del crecimiento de funciones. Aproximación de sumas por integrales. Estimación de soluciones de desigualdades recurrentes.
- Conocimiento detallado de algoritmos básicos de ordenación (Inserción, Quicksort, Mergesort, Heapsort): Pseudocódigos, evolución detallada sobre ejemplos, evolución sobre árboles de decisión. Análisis de eficacia en casos peor, medio y mejor. Cotas inferiores de rendimiento para familias de algoritmos.
- Conocimiento detallado del funcionamiento de algoritmos de búsqueda sobre claves: Pseudocódigos, evolución detallada sobre ejemplos. Tipo de datos diccionario: implementación sobre árboles binarios de búsqueda y árboles AVL, rendimiento de primitivas.
- Construcción de funciones hash y resolución de colisiones por encadenamiento y direccionamiento abierto.
- Alcance de un nivel medio-alto de programación en C: trabajo con punteros y memoria dinámica, determinación y control de errores, organización de código, archivos de cabecera.

### 1.12.3. Objetivos de la asignatura

Los **objetivos generales** son que el estudiante al acabar el curso logre:

- Tener un buen conocimiento general de los principales algoritmos de ordenación y búsqueda
- Ser capaz de determinar el efecto que las estructuras de un algoritmo tienen en su rendimiento.
- Identificar para un algoritmo dado los elementos determinantes de su rendimiento en el caso peor.
- Ser capaz de escribir y resolver las ecuaciones que recogen el rendimiento de algoritmos simples, incluyendo ecuaciones recursivas simples.
- Determinar para un problema dado el coste y ventajas de posibles enfoques algorítmicos para el mismo.
- Comunicar de manera clara, estructurada y concisa los resultados de su trabajo.
- Participar activamente en los análisis y discusiones de grupo que se establezcan al hilo del desarrollo de las actividades propuestas, cooperar con otros compañeros en el desarrollo de trabajos conjuntos y comunicar con propiedad y corrección sus reflexiones sobre los resultados de su trabajo.

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	3/9	

## Objetivos específicos

Al terminar cada unidad el estudiante debe ser capaz de:

### Unidad 1. Introducción al análisis de eficacia de algoritmos:

- Enumerar las medidas básicas de eficacia.
- Efectuar un primer análisis de la eficacia de algoritmos simples.
- Manejar las notaciones  $O$ ,  $\Theta$  y similares de crecimiento asintótico de funciones.
- Estimar el crecimiento de funciones expresadas como sumas.
- Definir y aplicar el rendimiento en los casos peor, mejor y medio.

### Unidad 2. Algoritmos básicos de ordenación

- Evaluar el rendimiento del algoritmo de inserción y su eficacia.
- Definir algoritmo local y enunciar las cotas inferiores para sus casos peor y medio.
- Estimar el rendimiento de un algoritmo local para algunas tablas dadas.

### Unidad 3. Algoritmos avanzados de ordenación

- Evaluar el rendimiento de un algoritmo de tipo Divide y Vencerás y obtener la estructura general de las desigualdades recurrentes a partir de un pseudocódigo dado.
- Enunciar el pseudocódigo y rendimiento en los casos peor y medio del algoritmo mergesort.
- Reproducir el funcionamiento del algoritmo mergesort sobre una tabla dada.
- Enunciar el pseudocódigo y rendimiento en los casos peor y medio del algoritmo quicksort.
- Reproducir el funcionamiento del algoritmo quicksort sobre una tabla dada.
- Enunciar la definición de heap y max-heap.
- Representar un heap en forma de árbol binario y en forma de tabla.
- Enunciar el pseudocódigo y rendimiento en el caso peor del algoritmo heapsort.
- Reproducir el funcionamiento del algoritmo heapsort sobre una tabla dada.
- Reproducir los argumentos empleados en el cálculo de esos casos.

### Unidad 4. Árboles de decisión para algoritmos de ordenación.

- Definir árbol de decisión.
- Construir árboles de decisión sobre algoritmos concretos y tablas de tamaño 3 y 4.
- Estimar las cotas inferiores para algoritmos de ordenación por comparación de claves en los casos peor y medio.
- Reproducir los argumentos empleados en la obtención de dichas cotas.

### Unidad 5. Algoritmos y EdDs básicos de búsqueda

- Enunciar el concepto del TAD Diccionario
- Usar árboles binarios como EdD para el TAD Diccionario.
- Calcular los valores de los casos peor y medio para la búsqueda en árboles binarios.
- Enunciar la definición de AVL
- Construir un AVL a partir de una tabla determinada.
- Estimar el caso peor de un AVL.
- Reproducir los argumentos empleados en la obtención de dichas propiedades.

### Unidad 6. Tablas hash

- Enunciar la definición de función hash y los principales métodos de la resolución de colisiones.
- Construir una tabla hash a partir de una tabla determinada.
- Construir tablas hash mediante encadenamiento.
- Definir tablas hash mediante direccionamiento abierto y enunciar los principales métodos de sondeo.
- Calcular el caso medio con éxito de una tabla hash con direccionamiento abierto a partir del caso medio sin éxito

## 1.13. Contenidos del programa

El programa detallado del curso es:

1. Análisis de eficacia de algoritmos:

a. Tipos de análisis. Medidas de eficacia.

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	4/9	

- b. Herramientas matemáticas: notaciones  $o$ ,  $O$ , etc.
  - c. Crecimiento de sumas e integrales.
  - d. Casos peor, mejor y medio.
2. Algoritmos básicos de ordenación
- a. Algoritmos de selección y burbuja.
  - b. Algoritmo de inserción: pseudocódigo, eficacia.
  - c. Algoritmos locales: definición, cotas inferiores para sus casos peor y medio.
3. Algoritmos avanzados de ordenación.
- a. Métodos Divide y Vencerás.
  - b. Desigualdades recurrentes. Estimación del crecimiento.
  - c. Algoritmo mergesort: pseudocódigo, caso peor.
  - d. Algoritmo quicksort: pseudocódigo, caso peor, caso medio.
  - e. Algoritmo heapsort: pseudocódigo, caso peor.
4. Árboles de decisión para algoritmos de ordenación.
- a. Árboles de decisión: concepto y construcción
  - b. Cotas inferiores para algoritmos de ordenación por comparación de claves.
5. Algoritmos básicos de búsqueda
- a. Búsqueda lineal y binaria.
  - b. TAD Diccionario. Árboles binarios como EdD para diccionarios.
  - c. Caso peor y medio para la búsqueda en árboles binarios.
  - d. Árboles AVL: construcción y profundidad máxima.
6. Tablas hash
- a. Construcción de funciones hash y resolución de colisiones.
  - b. Tablas hash sobre encadenamiento.
  - c. Tablas hash con direccionamiento abierto: sondeos lineales, cuadráticos y aleatorios.

### 1.14. Referencias de consulta

No hay un manual que se ajuste en su totalidad a los contenidos del curso. En cualquier caso, las referencias inferiores constituyen un buen complemento para el seguimiento del curso.

Referencias básicas:

- [Weiss, Data structures and algorithm analysis in C, Benjamin Cummings.](#)

Referencias complementarias:

- [Cormen, Leiserson, Rivest, Introduction to algorithms, The MIT Press--Mc Graw Hill.](#)
- [Baase, Computer algorithms, Addison-Wesley.](#)

Referencia para programación

- [Kernighan, Ritchie, The C programming language, Prentice hall.](#)

## 2. Metodologías docentes y tiempo de trabajo del estudiante

### 2.1. Presencialidad

	#horas
Porcentaje de actividades presenciales (mínimo 33% del total)	71

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	5/9	

## 2.2. Relación de actividades formativas

Actividades presenciales	Nº horas
Clases teóricas en aula	25
Seminarios	
Clases prácticas en aula	12
Prácticas clínicas	
Prácticas con medios informáticos	24
Prácticas de campo	
Prácticas de laboratorio	
Prácticas externas y/o practicum	
Trabajos académicamente dirigidos	
Tutorías	4
Actividades de evaluación	6
Otras	

## 3. Sistemas de evaluación y porcentaje en la calificación final

### 3.1. Convocatoria ordinaria

La asignatura consta de una parte práctica y una teórica. Ambas partes se puntúan sobre 10 puntos. La nota final de la asignatura se obtiene a partir de las notas obtenidas en las partes de teoría y de prácticas como sigue

$$\text{Calificación} = 0.4 * \text{Prácticas} + 0.6 * \text{Teoría}$$

Para aprobar la asignatura es obligatorio obtener una nota mayor o igual a 5 puntos, tanto en la parte de teoría como en la práctica. En caso contrario, la nota final que figurará en actas será

$$\text{Calificación} = 0.4 * \text{Mín}(5, \text{Prácticas}) + 0.6 * \text{Mín}(5, \text{Teoría})$$

Las notas de teoría y de prácticas se guardan para la convocatoria extraordinaria del mismo curso académico.

### Evaluación en el itinerario de evaluación continua:

La parte de teoría del curso se evaluará en el itinerario de Evaluación Continua mediante tres exámenes parciales, correspondientes cada uno de ellos aproximadamente a una tercera parte del curso. Tendrán una duración de 1 hora con dos preguntas sobre 10 puntos de varios apartados. Dichos exámenes tendrán lugar de acuerdo al siguiente calendario aproximado

- Parcial 1: en la semana del 17 de octubre de 2017.
- Parcial 2: en la semana del 28 de noviembre de 2017.
- Parcial 3: primeros de enero de 2018, coincidiendo con la fecha del examen final.

Los parciales 1 y 2 serán liberatorios para aquellos estudiantes que los superen obteniendo además una nota de 5 o superior en cada pregunta.

Caso de no superar alguno de estos dos parciales, el estudiante se examinará de nuevo del mismo (o, en su caso, de ambos) en el examen final, además de, naturalmente, la tercera parte del curso.

La nota final de teoría se calculará como la media de las tres notas de cada parte del curso. Aquellos estudiantes que hayan liberado los dos parciales deberán tener una nota de 4 o superior en el examen de la tercera parte para superar la asignatura.

### Evaluación fuera del itinerario de evaluación continua:

Para aquellos estudiantes que no hayan seguido el itinerario de Evaluación Continua, la nota final será exclusivamente la del examen final de 180'. En este caso, dicho examen **tendrá una primera parte en la que será necesario obtener una nota mínima de 7 para superarlo.**

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	6/9	

**Observación importante:**

La nota de teoría sólo se guardará para la convocatoria extraordinaria en el caso de superarse la materia en su totalidad. **No se guardarán notas de parciales individuales.**

En la **convocatoria ordinaria**, habrá una **prueba final de prácticas** con dos modalidades, según el estudiante haya seguido o no el itinerario de evaluación continua.

**Evaluación en el itinerario de evaluación continua:**

En este itinerario la prueba será de una hora y supondrá efectuar una modificación en el laboratorio de alguna de las tres prácticas realizadas y la respuesta a algunas cuestiones sobre la misma.

**Evaluación fuera del itinerario de evaluación continua:**

En este caso el estudiante deberá haber completado y entregado previamente a la fecha del examen las prácticas propuestas y la prueba, de dos horas, supondrá efectuar en el laboratorio una modificación de cierto alcance de alguna de las prácticas propuestas así como responder a algunas cuestiones.

En ambos casos la calificación de la parte práctica de la asignatura será:

**40% Nota Examen + 20% Nota Pract1 + 20% Nota Pract2 + 20% Nota Pract3**

Para poder aplicar la fórmula anterior el estudiante deberá, en ambos itinerarios, obtener al menos una calificación de 4 puntos en cada una de las prácticas, un promedio de al menos 5 puntos entre las tres entregas y una calificación de al menos 5 puntos en el examen. Si el estudiante no cumple alguna de estas condiciones la calificación de prácticas se calculará mediante la siguiente fórmula:

$$0.4 * \min(5, \text{Nota examen}) + 0.2 * \min(5, \text{Nota Prac 1}) \\ + 0.2 * \min(5, \text{Nota Prac 2}) \\ + 0.2 * \min(5, \text{Nota Prac 3})$$

En caso de retraso en la entrega de las prácticas, se podrá imponer una penalización en la nota de la práctica, que en ningún caso será superior a un punto por semana de retraso.

La nota de prácticas podrá guardarse de un curso al siguiente **una única vez** y siempre que se haya superado una determinada nota. Dicha nota se indicará a los estudiantes en la primera clase teórica del curso siguiente para que los interesados procedan a solicitar la aplicación de la nota del año anterior.

En la **convocatoria extraordinaria** el estudiante es evaluado fuera del itinerario de evaluación continua, conservando, si procede, la calificación de las prácticas con una calificación superior a 4. El estudiante deberá entregar, si no lo ha hecho ya en la convocatoria ordinaria y previamente a la realización del examen de la convocatoria extraordinaria la totalidad de las prácticas y realizar el examen de prácticas correspondiente a la convocatoria extraordinaria.

**ATENCIÓN:**

Cualquier copia descubierta que se haya realizado a lo largo del curso, tanto en cualquiera de las actividades de teoría como en cualquiera de los apartados de las prácticas, penalizará por igual, tanto a los alumnos que copian como a los copiados. La penalización por copia implica la aplicación de la normativa interna. La penalización por copia implica la aplicación de las normativas de la UAM y interna de la EPS <http://www.ii.uam.es/esp/alumnos/informacion/normativa.php>

**3.1.1. Relación actividades de evaluación**

Actividad de evaluación	%
Examen final de teoría (si no se sigue el itinerario de evaluación continua).	60
Examen final de prácticas (si no se sigue el itinerario de evaluación continua)	40
Evaluación continua de teoría (3 exámenes intermedios)	60
Evaluación continua de prácticas (evaluación de 3 prácticas; examen de prácticas)	40

Código Seguro de Verificación:	Fecha:	14/01/2021
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva	
Url de Verificación:	Página:	7/9



### 3.2. Convocatoria extraordinaria

Examen de teoría sobre toda la materia.

Examen de prácticas.

#### 3.2.1. Relación actividades de evaluación

Actividad de evaluación	%
Examen final de teoría	60
Examen final de prácticas	40

### 4. Cronograma orientativo

Semana	Clase	Contenido
1	1	Tema I: Tiempo abstracto de ejecución (TAE)
	2	Ejemplos de cálculo, análisis de rendimiento
	3	OB, Ejemplos de cálculo
2	4	Comp. Asint. de funciones
	5	Est. Crecimiento I
	6	Est. Crecimiento II. Complejidad de algoritmos
3	7	Complejidad de algoritmos
	8	Complejidad de la búsqueda lineal
	9	Problemas (1)
4	10	Problemas (2)
	11	Problemas en grupo I
	12	Tema II: Mét. básicos de ordenación. Insertsort
5	13	Métodos locales de ordenación. Cotas inferiores caso peor
	14	Cotas inferiores caso medio.
	15	Problemas (3)
6	16	Problemas (4)
	17	Problemas (5)
	18	Métodos DyV, MergeSort
7	19	Parcial I
	20	Análisis de MergeSort. Introd a QuickSort
	21	QuickSort, caso medio
8	22	Problemas (6)
	23	HeapSort
	24	Árboles de decisión. Ejemplos
9	25	Cotas inf para el caso peor de ord. por cdc
	26	Cotas inf para el caso de ord. por cdc
	27	Problemas (7)
10	28	Tema III: Revisión de búsq. lineal, búsq. binaria
	29	TAD Diccionario: primitivas
	30	TAD Diccionario: implementación
11	31	Rend. de primitivas de diccionario sobre ABdB
	32	Árboles AVL: construcción: rotaciones, ejemplos
	33	Prof. Máx. de AVLs, rend. sobre AVLs
12	34	Problemas (8)
	35	Problemas (9)
	36	Parcial II

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	8/9	



13	37 Hashing: introducción
	38 Hashing: funciones hash
	39 Problemas (10)
14	40 Hashing por encadenamiento
	41 Hashing por direccionamiento abierto
	42 Caso medio de hashing por DA uniforme
15	43 Caso medio de hashing por DA lineal y cuadrático
	44 Problemas (11)
	45 Problemas (12)

Código Seguro de Verificación:		Fecha:	14/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	9/9	