



UNIVERSIDAD  
DE GRANADA

*Este documento está protegido por la Ley de Propiedad Intelectual ([Real Decreto Ley 1/1996 de 12 de abril](#)).  
Queda expresamente prohibido su uso o distribución sin autorización del autor.*

# Algorítmica

## 2º Grado en Ingeniería Informática

### Cómo aprobar la asignatura

Descripción de planificación, tareas y evaluación

1. Descripción del documento.....	2
2. Resumen.....	2
3. Contenidos del curso.....	2
4. Evaluación.....	3
5. Seguimiento de la asignatura.....	4
6. Planificación.....	5
7. Seguimiento semipresencial u online de la asignatura.....	6

© Prof. Manuel Pegalajar Cuéllar  
Dpto. Ciencias de la Computación e I. A.  
Universidad de Granada



**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

# Cómo aprobar la asignatura

## 1. Descripción del documento

Este documento pretende ser una guía de la planificación de la asignatura *Algorítmica*. En él, se presentan los contenidos a impartir, organizados por temas, así como la planificación a seguir durante el curso, las tareas a realizar y su evaluación.

## 2. Resumen

- **Contenidos de la asignatura:** Organización en 5 temas o módulos:
  - La eficiencia de los algoritmos (3 semanas)
  - Algoritmos Divide y Vencerás (3 semanas)
  - Algoritmos voraces (3 semanas)
  - Algoritmos de Programación Dinámica (2 semana)
  - Algoritmos de exploración en grafos (2 semanas)
- **Evaluación en escenario A:** Hay 2 bloques susceptibles de ser evaluados: Teoría (**hasta 70% de la calificación final**) y prácticas (**desde 30% de la evaluación final**). Se presentan 2 modalidades para la evaluación:
  - **Evaluación continua:**
    - **Entrega de prácticas (desde 30%)**
    - **Evaluación de prueba final de conocimientos (hasta 70%)**
  - **Evaluación única final:** La evaluación se realizará en un único acto académico en la fecha establecida por el Centro y consistirá en un examen (evaluado de 0 a 10) que incluirá preguntas tanto de tipo teórico como práctico que garanticen que el alumno ha adquirido la totalidad de las competencias descritas en la guía docente.
  - **Convocatoria extraordinaria:** Se aplican los mismos criterios que para la evaluación única final. En caso de haber superado las prácticas, se podrá mantener la calificación obtenida.
- **Evaluación en escenario B:** La calificación final se obtiene como desde 70% prácticas y hasta 30% teoría (ver guía docente para ampliar información). **LA ENTREGA DE TODAS LAS PRÁCTICAS ES OBLIGATORIA EN EL ESCENARIO B.**
- **Planificación:** Cada módulo/tema se imparte en un rango temporal entre 2 y 3 semanas. Al finalizar, se realizarán ejercicios de clase para resolver problemas con diferentes técnicas estudiadas en la asignatura (2 semanas).

## 3. Contenidos del curso

La asignatura tiene un total de 6 créditos ECTS, de los cuales 4.5 son teóricos y 1.5 prácticos. En

total, se organizan en 60 horas de trabajo presencial y 90 horas de trabajo autónomo del estudiante. Los contenidos, ampliados de la guía docente, se organizan en 5 módulos, más uno final de ampliación:

Número de módulo	Módulo	Contenidos
1	La eficiencia de los algoritmos	Se describen técnicas para el estudio y análisis de eficiencia de algoritmos, tanto iterativos como recursivos. Se explica el concepto de orden de eficiencia y los diferentes tipos de órdenes.
2	Divide y Vencerás	Se explica la metodología de diseño de algoritmos Divide y Vencerás, y se exponen ejemplos y ejercicios de resolución de diferentes problemas resueltos con esta técnica.
3	Greedy	Se explica la metodología de diseño de algoritmos voraces (Greedy), y se exponen ejemplos y ejercicios de resolución de diferentes problemas resueltos con esta técnica.
4	Programación Dinámica	Se explica la metodología de diseño de algoritmos de Programación Dinámica, y se exponen ejemplos y ejercicios de resolución de diferentes problemas resueltos con esta técnica.
5	Exploración en grafos	Se explican diferentes metodologías de diseño de algoritmos para resolución de problemas que pueden plantearse como una búsqueda en espacios de estados, y se exponen ejemplos y ejercicios de resolución de diferentes problemas resueltos con estas técnicas.
5	Ampliación	Se introducen metodologías de resolución de problemas utilizando técnicas de diseño de algoritmos avanzadas.

## 4. Evaluación

La evaluación de la asignatura se divide en dos bloques: Teoría y Prácticas. Para el cálculo de la calificación final en el escenario A, el bloque de teoría se ponderará hasta 70%, y el bloque de prácticas se ponderará desde 30%. En el escenario B, la evaluación se realizará mediante desde 70% (ponderación de prácticas) y hasta 30% (ponderación de teoría). La calificación final será un valor numérico expresado de 0 a 10. **LA ENTREGA DE TODAS LAS PRÁCTICAS ES OBLIGATORIA EN EL ESCENARIO B.**

En el escenario actual:

- 20% de cada práctica para la calificación final (4 prácticas en total).
- 20% examen final.

## 4.1. Evaluación única final y Evaluación extraordinaria

Para la evaluación única final, y también para la convocatoria extraordinaria, se propondrá al estudiante un examen con preguntas de índole teórico y práctico, que serán calificadas de 0 a 10.

## 4.2. Carácter de no presentado

Un estudiante será calificado en cada convocatoria como **no presentado** si:

- Opta por evaluación continua y no ha entregado prácticas, o no se presenta al examen final.
- Opta por evaluación única final y no se presenta al examen.
- No se presenta al examen en la convocatoria extraordinaria.

## 5. Seguimiento de la asignatura

La metodología a seguir en la asignatura será la metodología de clase invertida (flipped classroom). En esta metodología se seguirán los siguientes pasos:

1. **Con carácter previo a la docencia presencial**, el profesor pondrá material de estudio a disposición del estudiante, para que realice el trabajo de aprendizaje (videos, lecturas, etc.). También, con carácter semanal, se proporcionará una guía de estudio de los conceptos y tareas relacionadas con los módulos semanales estudiados.
2. **Las clases presenciales se utilizarán para:**
  1. Repasar los conceptos teóricos estudiados previamente por el estudiante.
  2. Resolución de dudas.
  3. Realización de ejercicios que refuercen los conceptos teóricos.
  4. Fortalecer y abordar en mayor profundidad los conceptos teóricos aprendidos.
  5. En las clases diseñadas para elaboración de prácticas: Realización de las tareas asignadas, presentación de dudas o resultados parciales al profesor para verificar la correcta ejecución de la tarea.
3. Semanalmente, se propondrá al estudiante la elaboración de apuntes (**libreta**), de modo que el estudiante pueda sintetizar los conceptos más relevantes estudiados en cada semana del curso.

Dado el carácter de la asignatura, se recomienda al estudiante el seguimiento semanal de la misma, mediante:

- El estudio del material propuesto (videos, diapositivas, lecturas).
- La elaboración de la libreta de apuntes.
- La elaboración de las tareas propuestas.

La consecución con éxito de la asignatura dependerá, en mayor medida, de acatar estas recomendaciones con firmeza por parte del estudiante.

## 6. Planificación

La planificación de horarios de clase dada por el centro organiza la asistencia presencial de forma semanal, con 4 horas de clase en cada semana, dando lugar a un total de 15 semanas de clase. Contando con 3 semanas dedicadas a evaluación (presentación de tareas, examen final -en su caso-, etc.), la exposición y estudio de contenidos se organizará en un total de 12 semanas. La planificación temporal por módulos es la siguiente, donde cada fila se corresponde con un módulo de estudio (ver tabla anterior), y cada columna se corresponde con semanas de clase. Las semanas marcadas de color rojo y texto "**EVA**" se corresponden con semanas de **inicio de evaluación** (presentación de tareas a realizar, evaluación de tareas en clase, entrega de prácticas, etc.).

Módulo	SEMANA														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	EVA										EVA	EVA
2				2	2	2	EVA							EVA	EVA
3							3	3	3	EVA				EVA	EVA
4										4	4		EVA	EVA	EVA
5												5	5	EVA	EVA

## 7. Seguimiento semipresencial u online de la asignatura

Con el fin de facilitar al estudiante el seguimiento no presencial u online de la asignatura, se han desarrollado un conjunto de videos explicando cada uno de los conceptos de cada tema de la asignatura. A continuación, se detalla el contenido de estos videos por tema y semana:

### 7.1. TEMA 1

#### SEMANA 1 (22F-28F) : 90' de videos

- **QueEsUnAlgoritmo (19' 07'')**. Explica la definición de algoritmo y sus propiedades. Se exponen ejemplos de casos que no son algoritmos, y el diseño de algoritmos válidos.
- **LaEficienciaDeLosAlgoritmos (9' 56'')**. Expone qué es el concepto de medición de la eficiencia de un algoritmo, los tipos de casos en los que es útil cuantificar la eficiencia de un procedimiento, y la notación usada para referirnos al análisis de un problema en términos de eficiencia. Se muestra un ejemplo para el caso del problema de ordenación, contabilizando el tiempo de ejecución de 3 algoritmos que resuelven el problema y verificando cuál de ellos es más eficiente, en términos intuitivos.

- **El Principio de Invarianza (17' 55'')**. El video introduce el Principio de Invarianza, y lo relaciona con los órdenes de eficiencia posteriores. Se estudia un ejemplo práctico de este principio con el algoritmo de Burbuja implementado en varios lenguajes de programación.
- **El Orden O (24' 15'')**. En el video se realiza una introducción a los órdenes de eficiencia, y en particular, al orden O. Se explica cómo poder comparar órdenes de eficiencia diferentes y, finalmente, calcular la constante oculta.
- **El Orden Omega (18' 15'')**. El video se centra en la descripción del orden de eficiencia Omega, la explicación del orden Theta, y las propiedades de los órdenes de eficiencia, particularmente en las propiedades del máximo, la suma y el producto. También se realiza un cálculo experimental de la constante oculta para el mejor caso, en el algoritmo de ordenación por Burbuja.

#### SEMANA 2 (1M-7M) : 58' de videos

- **ReglasDelCalculoDeEficiencia (37' 00'')**. El video muestra cómo analizar la eficiencia de un algoritmo experimentalmente, exponiendo las reglas del cálculo de eficiencia para operaciones elementales, sentencias condicionales, sentencias repetitivas, secuencias de sentencias, funciones y llamadas a funciones.
- **EjerciciosEficiencia (21' 02'')**. El video expone ejercicios resueltos sobre el cálculo de eficiencia de algoritmos iterativos.

#### SEMANA 3 (8M-14M): 119' de videos

- **AnálisisDeFuncionesRecursivas (18' 50'')**. El video explica cómo se calcula la eficiencia cuando tenemos que analizar un algoritmo recursivo. Establece el concepto de ecuación en recurrencias y expone cómo calcular la ecuación en recurrencias de la implementación de un algoritmo.
- **EcuacionesLinealesHomogeneas (20' 31'')**. El video explica la necesidad de resolver la ecuación en recurrencias de un algoritmo recursivo para poder calcular su orden de eficiencia, e introduce el método de la ecuación característica para resolverlo. Se comienza explicando la identificación de ecuaciones lineales homogéneas y su resolución.
- **EcuacionesLinealesNoHomogeneas (18' 54'')**. El video profundiza en la resolución de ecuaciones lineales para el cálculo de la eficiencia de algoritmos. En particular, se aborda la identificación de ecuaciones lineales no homogéneas de coeficientes constantes, y su resolución.
- **CambioVariableRango (25' 48'')**. El video explica cómo abordar los problemas de resolución de ecuaciones lineales para el cálculo de la eficiencia de algoritmos, en los casos en los que es necesario realizar un cambio de variable y/o de rango para transformar la ecuación a una ecuación lineal homogénea, o lineal no homogénea. Se exponen ejemplos de ambos casos.
- **EjerciciosRecurrencias (33' 47'')**. En este video se expone la resolución de diversos ejercicios de ecuaciones en recurrencias.

## 7.2. TEMA 2

#### SEMANA 4 (15M-21M) : 95' de videos

- **LaTecnicaDivideYVenceras (21' 25'')**. El video explica los fundamentos de la técnica de diseño de algoritmos Divide y Vencerás. Se abordan en mayor profundidad los requisitos de la técnica para poder ser aplicada, su diseño de componentes y la estructura básica de algoritmos Divide y Vencerás.
- **ElProblemaDeOrdenacionAlgoritmoMergeSort (27' 53'')**. En este video se explica el enunciado del problema de ordenación y se proporciona una solución al mismo, siguiendo la

metodología de diseño de Divide y Vencerás. El algoritmo resultante es conocido como MergeSort. Se exponen sus componentes, su diseño y un ejemplo de ejecución del mismo.

- **ReduccionPorDyVProblemaExponenciacion (13' 57")**. El video introduce el caso particular de Divide y Vencerás cuando el problema se divide en un único problema de tamaño menor: La técnica de reducción. Se aplica a un problema de ejemplo: El problema de exponenciación. Se proporciona el diseño de dicho problema, su implementación, y un estudio de su eficiencia.
- **ElProblemaDelUmbral (13' 01")**. En este video se explica qué es el problema del umbral en la técnica Divide y Vencerás, y cómo resolverlo considerando las herramientas de cálculo de eficiencia disponibles. Finalmente, se expone un ejemplo de su aplicación en el problema de búsqueda, particularmente en la combinación del algoritmo MergeSort y el algoritmo de Inserción.
- **ElProblemaDeBusquedaDeElementosEnUnVector (18' 01")**, El video explica el problema de búsqueda en un vector, y una solución Divide y Vencerás para resolverlo. Como resultado, se obtiene el conocido algoritmo de Búsqueda Binaria. Se resuelve identificando las componentes de diseño del algoritmo, estableciendo la estructura del mismo y calculando la eficiencia con respecto al método básico. Finalmente, se expone un ejemplo de su funcionamiento.

#### SEMANA 5 (22M-28M) : 114' de videos

- **ElProblemaDeSeleccion (24' 47")**. El video introduce el problema de selección, proporcionando una solución mediante la técnica de reducción por divide y vencerás. Se diseñan las componentes del algoritmo, se calcula su eficiencia y se discute el caso mejor y peor del problema para el algoritmo diseñado.
- **MultiplicacionDeEnteros (25' 16")**. En este video se aborda el problema de multiplicación de números enteros largos. Se aborda el problema diseñando todas sus componentes y se muestra una solución preliminar. La solución preliminar sirve como ejemplo de cómo no siempre cualquier diseño Divide y Vencerás mejora a un algoritmo básico inicial, mediante el cálculo de la eficiencia del método diseñado. Finalmente, se propone un diseño alternativo que mejora la eficiencia a los dos métodos propuestos inicialmente.
- **MultiplicaciónDeMatrices (29' 08")**. El video aborda el problema de la multiplicación de matrices, y propone una solución mediante la técnica DyV que no mejora al método básico. Posteriormente, se expone el método Divide y Vencerás de Strassen para mejorar la eficiencia.
- **LaLineaDelHorizonte (34' 01")**. El video explica el problema de la línea del horizonte, proporcionando un algoritmo básico para resolverlo y una solución Divide y Vencerás que mejora la eficiencia del mismo. Se realiza un diseño de componentes, la estructura del algoritmo, estudio de eficiencia y un ejemplo de ejecución.

#### SEMANA 6 (5A-11A) : 43' de videos→ Festivo el 5

- **OrdenacionPorQuickSort (42' 33")**. El video vuelve a revisar el problema de ordenación de elementos en un vector, esta vez proponiendo una alternativa al diseño Divide y Vencerás expuesta para el algoritmo MergeSort. El resultado del diseño es el algoritmo de ordenación QuickSort. Se realiza un estudio de la eficiencia en el mejor y en el peor caso, y se ilustra un ejemplo de ejecución.

### 7.3. TEMA 3

#### SEMANA 7 (12A-18A) : 88 ' de videos



- **IntroduccionGreedy (20' 26")**. El video introduce la técnica Greedy de diseño de algoritmos. En particular, se revisan sus ventajas, inconvenientes, y se proporciona un ejemplo de algoritmo Greedy sobre el problema del cambio de monedas que, con un diseño poco elegante, puede llegar a no dar la solución óptima a un problema o, en el peor caso, no encontrar ninguna solución aunque esta exista.
- **MetodologiaGreedy (13' 22")**. El video introduce las componentes de diseño greedy y la plantilla general del algoritmo. Muestra un ejemplo de diseño greedy haciendo uso del problema del cambio de monedas estudiado en el video anterior.
- **Conceptos sobre grafos (9' 15")**. El video explica conceptos fundamentales sobre grafos, que serán utilizados para resolver los problemas abordados en los siguientes videos del tema.
- **AlgoritmoDeKruskal (27' 28")**. El video explica el problema del árbol generador minimal, y su resolución mediante el algoritmo de Kruskal. Se realiza un análisis inicial del problema, se diseñan las componentes greedy, se explica la ejecución del algoritmo, y se proporciona una idea intuitiva de la demostración de optimalidad del algoritmo.
- **AlgoritmoDePrim (17' 47")**. El video muestra otra alternativa de diseño Greedy para resolver el problema del árbol generador minimal, dando como resultado al algoritmo de Prim. Tras realizar el análisis del problema, el diseño de componentes y la estructura del algoritmo, se proporciona una idea básica intuitiva de la demostración de optimalidad del mismo, y se muestran algunas aplicaciones del problema como consideraciones finales.

#### SEMANA 8 (19A-25A) : 82' de videos

- **CaminosMinimos (45' 45")**. El video explica un problema de caminos mínimos sobre un grafo, y su solución mediante la técnica Greedy. Al algoritmo resultante se le conoce como Algoritmo de Dijkstra. Tras realizar el diseño del mismo, mostrar su estructura y demostrar su optimalidad, se proporciona un ejemplo de su funcionamiento.
- **ComplejidadYHeurísticas (7' 33")**. El video realiza una descripción de problemas en términos de su complejidad, introduciendo el concepto de clases de complejidad P y NP. Seguidamente, introduce el concepto de heurística como método simple para discriminar entre buenas y malas decisiones, lo cual ayudará al diseño de algoritmos para optimización.
- **ProblemaColoreoGrafo (16' 15")**. El video muestra el problema del coloreo de un grafo, y proporciona dos posibles soluciones al mismo. Se estudia el ejemplo de funcionamiento de una de ellas, y se ilustra la demostración de no optimalidad mediante contraejemplo.
- **El problema de Lobezo y creación de equipos de super-héroes, planteado como un problema de Coloreo de un grafo (13' 10")**. En el enlace siguiente se muestra una aplicación del problema de coloreo de un grafo para ayudar a superhéroes a luchar contra supervillanos (<https://www.youtube.com/watch?v=59ajcwo7aw>) .

#### SEMANA 9 (26A-2M) : 107' de videos

- **ViajanteDeComercio (18' 33")**. El video introduce uno de los problemas clásicos de la algorítmica: El viajante de comercio. Se proporciona una solución greedy al problema, justificando sus componentes de diseño. Se visualiza el funcionamiento del algoritmo y, finalmente, se proporciona la demostración de no optimalidad.
- **MochilaContinuo (41' 29")**. El video expone el problema de la mochila y algunas de sus variantes. Se centra en el problema de la mochila continuo, y la propuesta de varias heurísticas para abordarlo. Se demuestra la optimalidad de la heurística óptima para este problema, dando también un ejemplo de funcionamiento del algoritmo resultante.
- **PlanificacionTareasSistema (25' 00")**. El video expone el problema de planificación de tareas en el sistema, dando una solución greedy óptima.
- **ProblemaAsignacionTareas (22' 05")**. El video explica el problema de asignación de n tareas a n trabajadores, con una correspondencia 1:1 (una tarea a cada trabajador y viceversa). El ejercicio muestra diversas alternativas para resolver el problema, cada una dando lugar a una solución distinta. También se demuestra la no optimalidad de ninguna de las soluciones indicadas, aunque se indica una referencia a un algoritmo que resuelve este problema particular de forma óptima: El algoritmo húngaro.



## 7.4. TEMA 4

### SEMANA 10 (3M-9M) : 89' de videos

- **ProgramacionDinamica (36' 07")**. El video introduce la técnica de Programación Dinámica, comparando similitudes y diferencias con las técnicas Divide y Vencerás y Greedy, estudiadas hasta el momento. También enuncia algunas características de diseño de algoritmos de Programación Dinámica, y explica cómo influyen estas características en dicho diseño mediante problemas de ejemplo que, sin llegar a ser problemas resolubles con Programación Dinámica, sí comparten similitudes que serán de utilidad para entender la técnica.
- **MetodologiaProgramacionDinamica (5' 00")**. El video establece los requisitos que se deben cumplir por un problema para que éste pueda ser resuelto mediante la técnica de Programación Dinámica. A continuación, explica las fases o etapas de diseño que hay que seguir hasta obtener el algoritmo resultante.
- **ProblemaCambioMonedas (47' 22")**, El video revisa el problema del cambio de monedas estudiado en el tema de Algoritmos Greedy. Se plantea una solución al mismo con la técnica de Programación Dinámica, diseñando todas sus componentes y dando una implementación de ejemplo.

### SEMANA 11 (10M-16M) : 137' de videos

- **ProblemaMochila0-1 (48' 30")**. El video plantea el problema de la mochila en su forma de Mochila 0/1, y describe una solución con Programación Dinámica al mismo. Se describe el análisis de componentes de diseño, la estructura del algoritmo, idea intuitiva de la demostración de optimalidad, y ejemplo de recuperación de la solución partiendo de la ecuación en recurrencias.
- **CaminosMinimosPD (43' 08")**. El video aborda otro problema de caminos mínimos, alternativo al estudiado previamente en la técnica de diseño Greedy. Como resultado, se diseña un algoritmo de Programación Dinámica para resolverlo, exponiendo cada una de sus componentes y comprobando su funcionamiento sobre una instancia de problema de ejemplo. El algoritmo resultante se conoce como Algoritmo de Floyd.
- **MultiplicacionEncadenadaMatrices (44' 40")**. El video explica el problema de multiplicación encadenada de matrices, y cómo puede ser resuelto de forma óptima mediante la técnica de programación dinámica.

## 7.5. TEMA 5

### SEMANA 12 (17M-23M) : 118' de videos.

- **TecnicasExploracionGrafos (30' 02")**. El video introduce las técnicas de exploración en grafos como mecanismo para resolver problemas. En particular, se diferencia entre grafo explícito y grafo implícito, se realiza un repaso de las técnicas básicas de exploración sobre grafos explícitos (anchura y profundidad), y se explica cómo modelar un espacio de estados como paso previo a la resolución de un problema.
- **MetodologiaBacktracking (17' 53")**. El video expone las componentes de diseño de la metodología de diseño de algoritmos Backtracking, así como la estructura básica del algoritmo diseñado con esta técnica, y su adaptación a posibles escenarios alternativos dependientes de la función objetivo a perseguir.

- **Problema8Reinas (32' 37")**. El video expone el problema de las N reinas, y proporciona un diseño de algoritmo backtracking para solucionarlo.
- **ResolucionSudoku (23' 45")**. El video muestra cómo diseñar un algoritmo con la técnica Backtracking para resolver Sudokus.
- **ViajanteComercioBT (12' 26")**. El video muestra cómo utilizar la técnica Backtracking para resolver problemas de optimización. En particular, se aborda el diseño de una solución para resolver el problema del Viajante de Comercio, devolviendo la solución óptima a este problema.

**SEMANA 13 (24M-30M) : 83' de videos.**

- **MetodologiaBranchAndBound (28' 33")**. El video expone la metodología Branch And Bound para diseño de algoritmos de exploración en grafos, la estructura general de su algoritmo y un ejemplo de la dinámica de ejecución del mismo.
- **AsignacionTareasBB (29' 44")**. El video expone un problema de optimización para asignar n tareas a n trabajadores, y su resolución mediante la técnica de exploración en grafos por ramificación y poda.
- **ViajanteComercioBB (24' 13")**. El video revisa el problema del viajante de comercio, y expone una alternativa de solución mediante la técnica de ramificación y poda. Se muestra un ejemplo de cómo funcionaría el esquema de algoritmo general de Branch&Bound para resolver una instancia del problema.

**SEMANA 14 (31M-6J)→ Festivo los 3-4**

- **Resolución de problemas**

**SEMANA 15 (7J-9J)**

- **Resolución de problemas**