



# Algoritmos y Estructuras de Datos II

Curso académico: 2014/2015

Titulación: Grado en Ingeniería Informática

Curso: 2º; Grupo: 1

Carácter: Obligatoria

Créditos: 6 (3 teóricos, 3 prácticos)

Profesor: Norberto Marín Pérez (teoría y prácticas)  
y José Ramón Hoyos (prácticas)



# Algoritmos y Estructuras de Datos II

Curso académico: 2014/2015

Titulación: Grado en Ingeniería Informática

Curso: 2º; Grupo: 3

Carácter: Obligatoria

Créditos: 6 (3 teóricos, 3 prácticos)

Profesores: Norberto Marín Pérez (teoría) y Domingo Giménez y José Ramón Hoyos (prácticas)

# Objetivos de la asignatura

## Objetivo central

SER CAPAZ DE ANALIZAR, COMPRENDER Y RESOLVER UNA AMPLIA VARIEDAD DE PROBLEMAS COMPUTACIONALES, DISEÑANDO E IMPLEMENTANDO SOLUCIONES EFICIENTES Y DE CALIDAD, COMO RESULTADO DE LA APLICACIÓN DE UN PROCESO METÓDICO

1. Resolución de problemas
2. Eficiencia y calidad
3. Proceso metódico

# Objetivos formativos

- Tomar conciencia de la importancia de realizar siempre un análisis y diseño previos del problema, como pasos anteriores a la implementación en un lenguaje de programación.
- Distinguir los distintos tiempos de ejecución que se utilizan en el estudio de algoritmos: caso más favorable, más desfavorable y promedio.
- Aprender las técnicas básicas del estudio teórico de algoritmos: análisis del tiempo por conteo de instrucciones y estudio de la ocupación de memoria. Comprender la importancia de este estudio.
- Aprender a realizar estudios experimentales, comprender su importancia y la necesidad de contrastar los resultados experimentales con los teóricos.

# Objetivos formativos

- Conocer las notaciones de complejidad  $O$  (o-grande, orden),  $\Omega$ ,  $\Theta$  y  $o$  (o-pequeña), diferenciando claramente el significado y utilidad de cada una.
- Saber comparar las complejidades de distintas funciones.
- Conocer la complejidad de las funciones que aparecen más frecuentemente en el estudio de algoritmos, así como la relación entre las complejidades de estas funciones.
- Conocer las técnicas básicas de resolución de ecuaciones de recurrencia: expansión de la recurrencia, método de la ecuación característica y utilización de fórmulas maestras.

# Objetivos formativos

- Comprender el papel de las condiciones iniciales (casos base) en la resolución de ecuaciones recurrentes, y saber determinar cuáles se deben aplicar para cada ecuación dada.
- Comprender la técnica de resolución por división en problemas más pequeños.
- Conocer y saber aplicar los esquemas básicos de los algoritmos divide y vencerás.
- Comprender la técnica de resolución de problemas por avance rápido y los distintos casos que se pueden presentar en la resolución de problemas por esta técnica: obtención de la solución óptima, de una solución no óptima, o no obtención de la solución.

# Objetivos formativos

- Entender el método de avance rápido como técnica para obtener una solución inicial a partir de la cuál se puede realizar una búsqueda local, y conocer algunos ejemplos que se pueden resolver de ese modo.
- Comprender la técnica de resolución de problemas por programación dinámica, e identificar las diferencias con divide y vencerás y con avance rápido.
- Entender la ventaja de la programación dinámica con respecto a otras técnicas en cuanto a tiempo de ejecución, al evitar la repetición de cálculos que aparece en métodos recursivos.
- Saber identificar problemas que cumplen el principio de optimalidad, que es necesario para poder aplicar esta técnica.

# Objetivos formativos

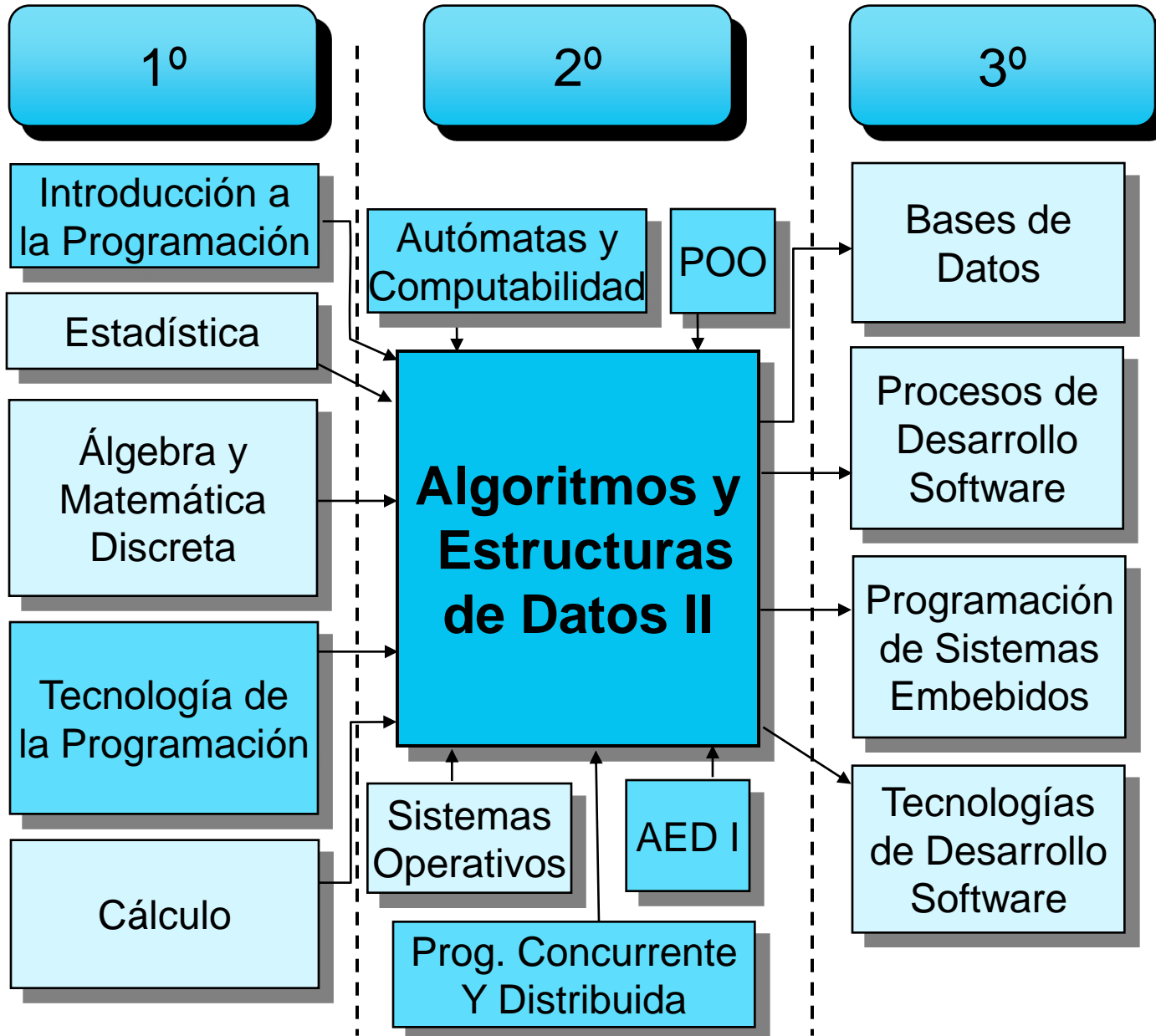
- Entender el concepto de árbol de solución, y su utilización en la resolución de problemas. Comprender que en muchos casos el árbol es sólo una representación lógica del conjunto de todas las posibles soluciones y subsoluciones.
- Comprender y saber aplicar la técnica de resolución de problemas por backtracking, entendiendo sus características principales y el concepto de búsqueda en profundidad.
- Comprender y saber aplicar la técnica de resolución de problemas por ramificación y poda, entendiendo la importancia de la estimación del beneficio para guiar la búsqueda, y del cálculo de las cotas para podar el árbol de búsqueda.
- Comprender la importancia de la heurística en la resolución eficiente de problemas de alto coste computacional.



# Competencias transversales

- Ser capaz de expresarse correctamente en español en su ámbito disciplinar.
- Ser capaz de gestionar la información y el conocimiento en su ámbito disciplinar, incluyendo saber utilizar como usuario las herramientas básicas en TIC.
- Ser capaz de trabajar en equipo y para relacionarse con otras personas del mismo o distinto ámbito profesional.

# Contexto curricular



**Plan Grado  
I.I. de 2009**

# Programa

## Algoritmos y Estructuras de Datos II

0. Introducción

1. Análisis de algoritmos

**Bloque I**

2. Divide y vencerás

3. Algoritmos voraces

4. Programación dinámica

5. Backtracking

6. Ramificación y poda

**Bloque II**

•Horarios de teoría: miércoles: 9 a 11, aula A03

•Horarios de laboratorio:

→ Subgrupo 1.2: martes, 12:20 a 14:00, laboratorio 1.5

→ Subgrupo 1.1: miércoles, 12:20 a 14:00 laboratorio 1.4

→ Subgrupo 1.3: jueves, 9:00 a 10:40 laboratorio 1.6

# Programa

## Algoritmos y Estructuras de Datos II

0. Introducción

1. Análisis de algoritmos

**Bloque I**

2. Divide y vencerás

3. Algoritmos voraces

4. Programación dinámica

5. Backtracking

6. Ramificación y poda

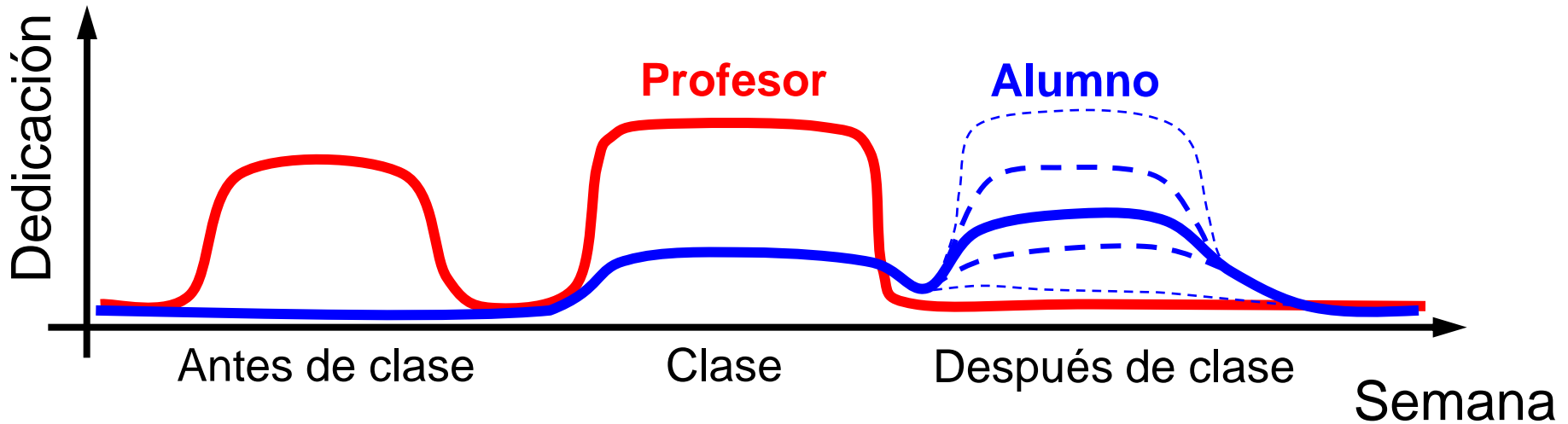
**Bloque II**

- Horarios de teoría: martes y miércoles: 17:30 a 18:30, aula A03
- Horarios de laboratorio:
  - Subgrupo 3.3: martes, 18:50 a 20:30, laboratorio 1.6
  - Subgrupo 3.2: miércoles, 18:50 a 20:30 laboratorio 1.5
  - Subgrupo 3.1: jueves, 18:50 a 20:30 laboratorio 1.4

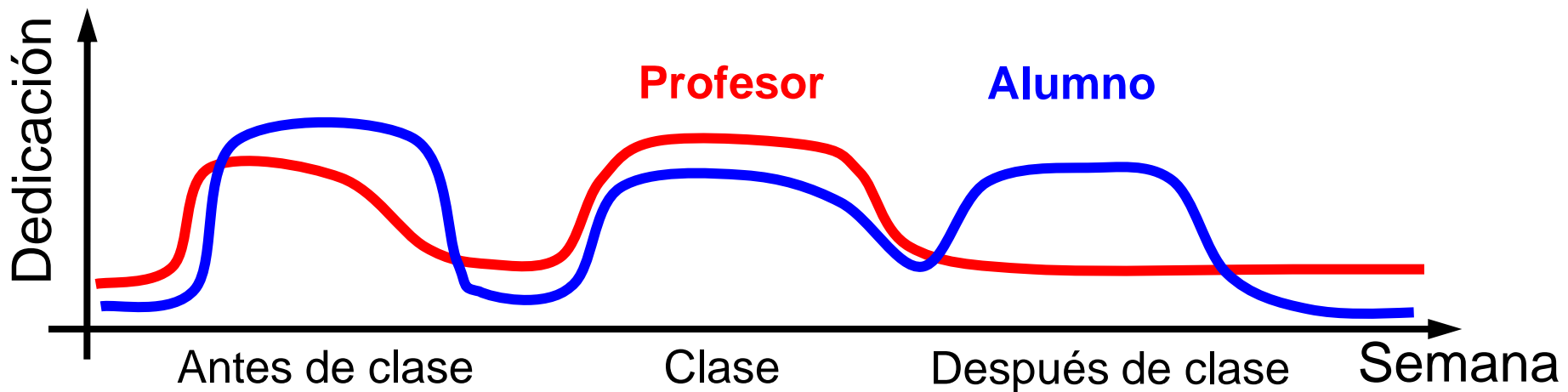
# El Problema con los Exámenes

## Espacio Europeo de Educación Superior (EEES)

- Modelo educativo anterior

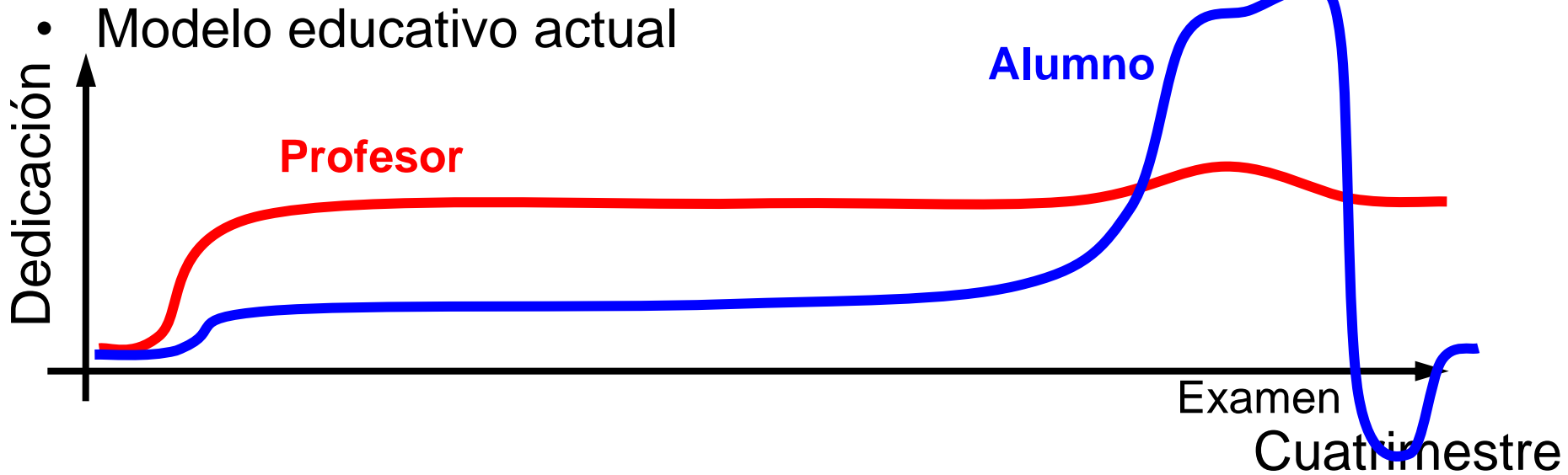


- Nuevo modelo educativo

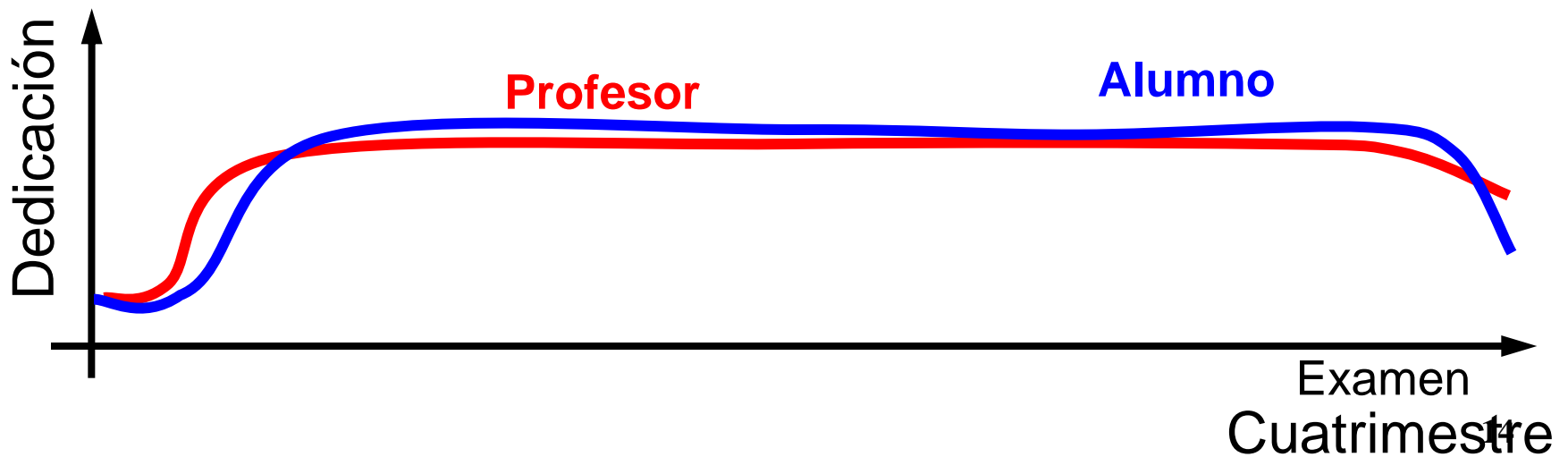


# El Problema con los Exámenes

## Espacio Europeo de Educación Superior (EEES)



- Nuevo modelo educativo



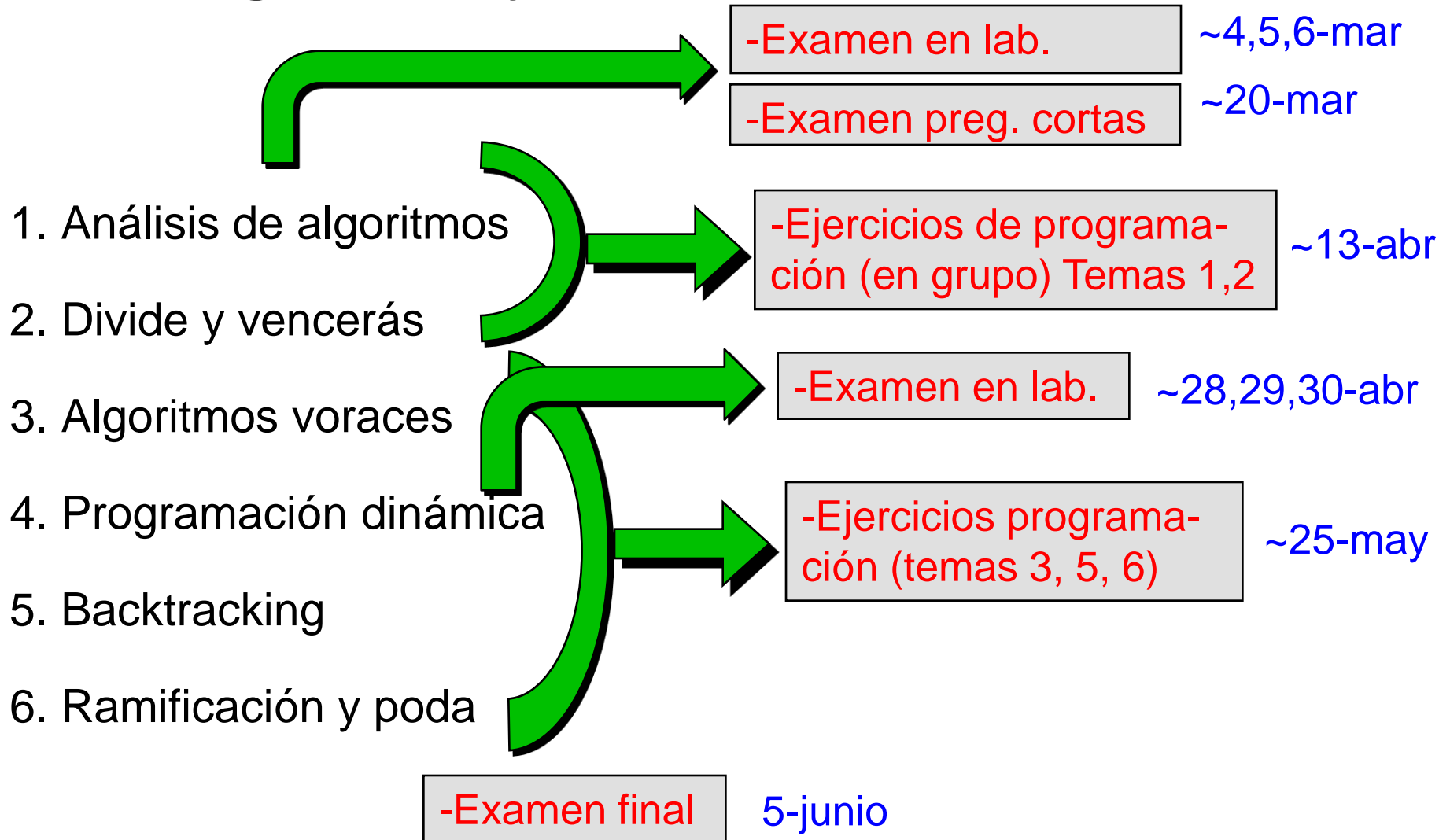
# Evaluación

## Principios

- **Examen del tema 1 y examen del tema 3**
- **Examen de tipo preguntas cortas (tema 1):** tiempos de ejecución, complejidad, ecuaciones de recurrencia...
- **Dos prácticas:** la primera sobre los temas 1 y 2, y la segunda sobre los temas 3, 5 y 6.
- **Examen final:** de todos los temas, excepto el tema 1 y el 4 si se han superado en el examen de preguntas cortas.
- Y por supuesto... **¡¡El juez on-line!!** Temas 2, 3, 5 y 6: resolución de problemas con diferentes técnicas algorítmicas: lenguajes C/C++, sobre Linux.

# Actividades de Evaluación Continua

## Algoritmos y Estructuras de Datos II





# Otras actividades...

- **Notas adicionales:**

*(+0,5 puntos sobre la nota final, siempre que esté aprobada la asignatura, en cada apartado)*

- Participación en clase (hasta 0,5)
- *Realización del curso de ACM (CPE) \*\*\**
- *Concurso de programación ACM Contest \*\*\**
- Otros...

# Mooshak: <http://dis.um.es/~mooshak>

Mooshak - Windows Internet Explorer

http://dis.um.es/~mooshak/cgi-bin/execute/632769135587

Mooshak

AED: Seminario de Maude

Gines Garcia Mateos

Mooshak

Problema

Programa

Listados

[más...](#)

101 - Naturales basicos

101 - Naturales basicos

102 - Operaciones multiplicativas

103 - Sustracciones

104 - Comparaciones

105 - Divisiones

106 - Operaciones avanzadas

107 - Naturales completos

110 - Vocales

111 - Pilas de vocales

112 - Pilas avanzadas

113 - Colas de vocales

114 - Colas avanzadas

115 - Listas de vocales

116 - Acceso a listas

117 - Modificadores de listas

118 - Consultas sobre listas

120 - Conjuntos de vocales

121 - Aritmetica de conjuntos

122 - Comparacion de conjuntos

125 - Bolsas de vocales

126 - Bolsas avanzadas

140 - Arboles binarios de vocales

141 - Arboles binarios avanzados

142 - Recorridos en arboles

143 - De arboles a bolsas

144 - Arboles balanceados

Ver

Preguntar

Enviar

Imprimir

Preguntas  Impresiones

con 15

Ayuda

Salir

Concurso terminado

PID: 101

Enunciado del problema 101

El Problema

Para comenzar esta serie de ejercicios vamos a trabajar con un tipo de datos elemental: los naturales. Los naturales son tipos de datos básicos, que se pueden definir en Maude. Damos por supuesto que el cero es un número natural. De esta forma, todos los naturales se pueden obtener a partir de la constante **cero** y la operación **sucesor**.

En definitiva, se pide escribir una especificación formal algebraica en Maude del tipo abstracto **Natural**, definida a través de los constructores **cero** y **sucesor**. La especificación debe usar el tipo **Real** y...

# Mooshak: <http://dis.um.es/~mooshak>

Mooshak - Windows Internet Explorer

http://dis.um.es/~mooshak/cgi-bin/execute/632769135587

Mooshak

Home Feeds (J) Print Page Tools

**AED: Seminario de Maude**  
Gines Garcia Mateos  
shak

**Problema** 105 - Divisiones

**Programa**

**Listados**  Envíos  Clasificación  Preguntas  Impresiones

[más...](#) Actualizar cada  minutos con  líneas

**Enunciado del problema 105 : Divisiones** **Concurso terminado**

**Divisiones**

PID: 105

**El Problema**

El **razonamiento inductivo** es una herramienta de una extraordinaria potencia. Esto explica que --incluso con una sintaxis tan reducida como la del lenguaje Maude-- cualquier operación, por compleja que sea, puede ser especificada convenientemente. No obstante, la eficiencia computacional obtenida para algunas operaciones será muy reducida; aunque tampoco es un problema que nos ocupe cuando estamos trabajando a nivel de especificación.

En este ejercicio vamos a definir operaciones relacionadas con la división entera entre dos números, trabajando con la especificación formal algebraica del tipo abstracto **Natural** del ejercicio 101. En

# Mooshak: <http://dis.um.es/~mooshak>

Mooshak - Windows Internet Explorer

http://dis.um.es/~mooshak/cgi-bin/execute/632769135587

Mooshak

**AED: Seminario de Maude**  
Gines Garcia Mateos  
shak

**Problema** 101 - Naturales basicos

**Programa**

**Listados**  Envíos  Clasificación  Preguntas  Impresiones

Actualizar cada  minutos con  líneas

[más...](#)

**Envíos** **Concurso terminado**

#	Tiempo Absoluto	País	Equipo	Problema	Lenguaje	Resultado	Estado
4085	2007/10/24 23:59		ITIS Gines Aroca Ruiz	106	Maude	Runtime Error	final
4084	2007/10/24 23:58		ITIS Jose Torrente Marchante	122	Maude	Wrong Answer	final
4083	2007/10/24 23:58		ITIS Jorge Salmeeron Fuentes	141	Maude	Runtime Error	final
4082	2007/10/24 23:58		ITIG Amalia Carrillo Sarabia	141	Maude	Runtime Error	final
4081	2007/10/24 23:58		ITIG Alejandro Rodriguez Yepes	141	Maude	Runtime Error	final
4080	2007/10/24 23:57		ITIG Amalia Carrillo Sarabia	141	Maude	Runtime Error	final
4079	2007/10/24 23:57		ITIS Jose Torrente Marchante	141	Maude	Runtime Error	final
4078	2007/10/24 23:57		ITIS Jorge Salmeeron Fuentes	141	Maude	Runtime Error	final



# AC (AntiCopias v1.7)

The screenshot displays the AC (AntiCopias v1.7) software interface. The main window shows a file explorer with 'AntiCopias v1.7 r211' selected. A 'Bzip2\_ncd\_sim' window displays a network graph with nodes 1, 2, 3, 8, 10, and 28. Below the graph is a histogram showing a distribution of values. Two code editors are open, showing C++ code for image processing. The left editor is '3:Unit2.08s.jass.amn.cpp' and the right is '1:Unit2\_08s\_mael\_cmmp.cpp'. Both editors show code for creating IplImage objects, converting RGB to YUV, splitting channels, and calculating averages for u and v channels.

```
3:Unit2.08s.jass.amn.cpp
IplImage *y=cvCreateImage(cvSize(yuv->w
IplImage *u=cvCreateImage(cvSize(yuv->w
IplImage *v=cvCreateImage(cvSize(yuv->w
IplImage *img=cvCreateImage(cvSize(yuv-

//Convertimos la imagen a yuv
iplRGB2YUV(foto[nfoto].img, yuv);

//Separamos los canales para operar con
cvSplit(yuv, y, u, v, NULL);
float sum_u=0;
float sum_v=0;
for (int i=0; i<u->height; i++){
    for(int j=0; j<u->width; j++){
        sum_u=sum_u+cvGet2D(u, i, j).val[
        sum_v=sum_v+cvGet2D(v, i, j).val[
    }
}
int mediau=sum_u/(u->width * u->height)
int mediav=sum_v/(u->width * u->height)

1:Unit2_08s_mael_cmmp.cpp
IplImage *y=cvCreateImage(cvSize(yuv->w
IplImage *u=cvCreateImage(cvSize(yuv->w
IplImage *v=cvCreateImage(cvSize(yuv->w
IplImage *img=cvCreateImage(cvSize(yuv-

//Convertimos la imagen a yuv
iplRGB2YUV(foto[nfoto].img, yuv);

//Separamos los canales para operar con
cvSplit(yuv, y, u, v, NULL);
float suma_u=0;
float suma_v=0;
for (int i=0; i<u->height; i++){
    for(int j=0; j<u->width; j++){
        suma_u=suma_u+cvGet2D(u, i, j).v
        suma_v=suma_v+cvGet2D(v, i, j).v
    }
}
int media_u=suma_u/(u->width * u->hei
int media_v=suma_v/(u->width * u->hei
```

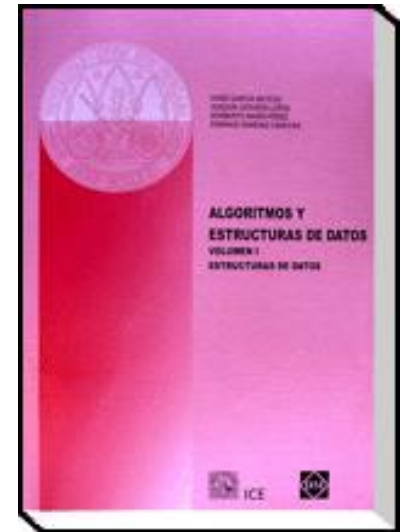
# Tutorías

- **Martes, de 11:15 a 12:15 y de 16:30 a 17:30**
- **Miércoles, de 11:15 a 12:15 y de 16:30 a 17:30**
- **Jueves, de 10:45 a 12:45**
  
- **Despacho 2.27 (2ª planta Fac. Informática)**
- **E-mail:** nmarin@um.es
- **Web asignatura:** <http://dis.um.es/~nmarin/>

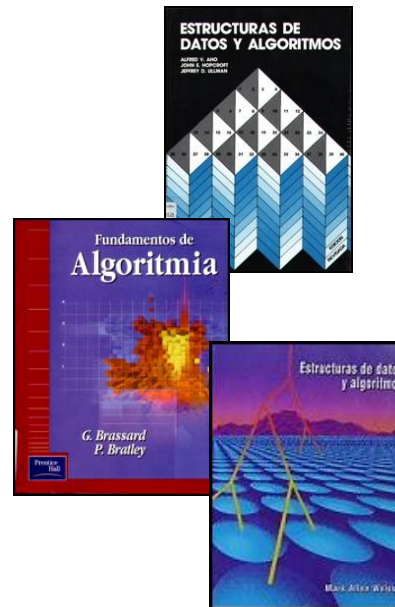
# Bibliografía

- **Algoritmos y Estructuras de Datos (texto guía)  
Volumen I y II**

N. Marín Pérez, G. García Mateos,  
D. Giménez Cánovas, J. Cervera López,  
Ed. Diego Marín, 2003



- **Estructuras de datos y algoritmos**  
A.C. Aho, J.E. Hopcroft, J.D. Ullman  
Addison-Wesley Iberoamericana, 1988
- **Fundamentos de Algoritmia**  
G. Brassard, P. Bratley  
Prentice-Hall, 1998
- **Estructuras de datos y algoritmos**  
Mark Allen Weiss  
Addison-Wesley Iberoamericana, 1995



(más en la web de la asignatura)