



# Algoritmos y Estructuras de Datos I

Curso académico: 2019/2020

Titulación: Grado en Ingeniería Informática

Curso: 2º; Grupo: 1 + PCEO

Carácter: Obligatoria

Créditos: 6 (3 teóricos, 0,75 seminarios, 2,25 prácticos)

Profesores: Norberto Marín (teoría)

Ginés García, Juan Manuel Carrillo,  
otro y Norberto Marín (prácticas)



# Algoritmos y Estructuras de Datos I

Curso académico: 2019/2020

Titulación: Grado en Ingeniería Informática

Curso: 2º; Grupo: 2

Carácter: Obligatoria

Créditos: 6 (3 teóricos, 0,75 seminarios, 2,25 prácticos)

Profesores: Norberto Marín (teoría)

Ginés García, Juan Manuel Carrillo,  
otro y Norberto Marín (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

- Entender el desarrollo de programas como un proceso metódico e ingenieril, formado por una serie de etapas con distintos niveles de abstracción, frente a la idea de la programación como arte.
- Reconocer la importancia de la abstracción y conocer los tipos de abstracciones que aparecen en programación: funcional, de datos y de iteradores.
- Concienciarse de la utilidad de desarrollar especificaciones completas y precisas, entendiendo la especificación como un punto de acuerdo entre el usuario y el implementador de una abstracción.

# Objetivos formativos

- Comprender el método de especificación formal algebraico o axiomático (basado en una definición mediante axiomas) y el método constructivo u operacional (basado en el uso de precondiciones y postcondiciones).
- Conocer la importancia y ubicuidad de los tipos conjunto y diccionario en el desarrollo de programas, independientemente de la estructura que se use para implementarlos.
- Ser capaz de diseñar, implementar y analizar la eficiencia de las principales estructuras de representación no arbóreas para los tipos conjunto y diccionario, adaptando el diseño a las necesidades específicas de cada aplicación.

# Objetivos formativos

- Conocer la estructura de datos de tablas de dispersión, sus distintas variantes y los factores que influyen en su eficiencia y uso de memoria.
- Conocer y comprender una variedad de técnicas eficientes de representación de conjuntos y diccionarios mediante estructuras arbóreas.
- Adquirir la capacidad de evaluar las necesidades de representación de una aplicación específica, tomando decisiones justificadas sobre las estructuras de representación más adecuadas.
- Comprender la necesidad de usar mecanismos de equilibrado o balanceo para conseguir eficiencia en las representaciones arbóreas.

# Objetivos formativos

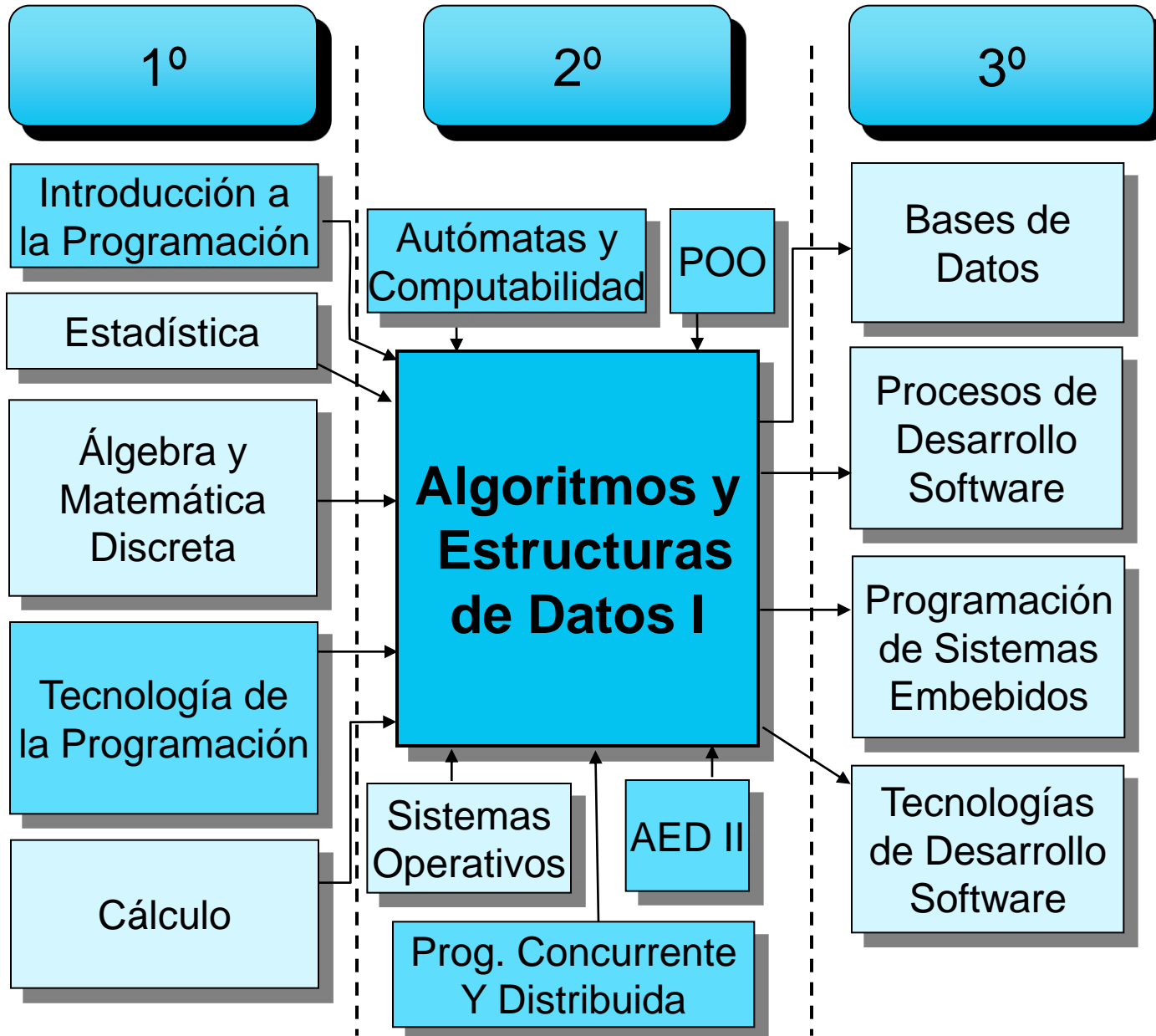
- Ser capaz de diseñar e implementar una estructura de datos para el tipo grafo –en sus distintas variantes– usando listas y matrices de adyacencia.
- Valorar críticamente las ventajas e inconvenientes de las representaciones de grafos mediante listas y matrices de adyacencia, y su influencia en la eficiencia de los algoritmos sobre grafos.
- Conocer y comprender el funcionamiento de una variedad de algoritmos clásicos sobre grafos (tales como los algoritmos de Prim, Kruskal, Dijkstra, Floyd y Warshall), razonando sobre las ideas subyacentes que aportan y analizando su complejidad computacional.
- Ser capaz de usar los algoritmos estudiados como herramientas prácticas para la resolución de problemas en un contexto genérico, a través de la transformación de un problema de interés en un problema sobre grafos.

# 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  
II de 2009**

# Programa

## Algoritmos y Estructuras de Datos I. Grupo 1+PCEO

|   |                   |
|---|-------------------|
| 0. Introducción                                 | <b>Bloque I</b>   |
| 1. Abstracciones y especificaciones             |                   |
| 2. Conjuntos y diccionarios                     | <b>Bloque II</b>  |
| 3. Representación de conjuntos mediante árboles |                   |
| 4. Grafos                                       | <b>Bloque III</b> |

- Horarios de teoría: martes, 9:00 a 11:00, aula A03
- Horarios de laboratorio:
  - Subgrupo 1: miércoles, 12:20 a 14:00, laboratorio 1.4
  - Subgrupo 2: jueves, 9:00 a 10:40, laboratorio 1.5
  - Subgrupo 3: jueves, 12:20 a 14:00, laboratorio 1.6
  - PCEO: miércoles, 17:10 a 18:50, laboratorio 2.2

# Programa

## Algoritmos y Estructuras de Datos I. Grupo 2

0. Introducción

1. Abstracciones y especificaciones

2. Conjuntos y diccionarios

3. Representación de conjuntos mediante árboles

4. Grafos

**Bloque I**

**Bloque II**

**Bloque III**

•Horarios de teoría: lunes, 9:25 a 11:25, aula A04

•Horarios de laboratorio:

→Subgrupo 1: jueves, 9:00 a 10:40, laboratorio 2.3

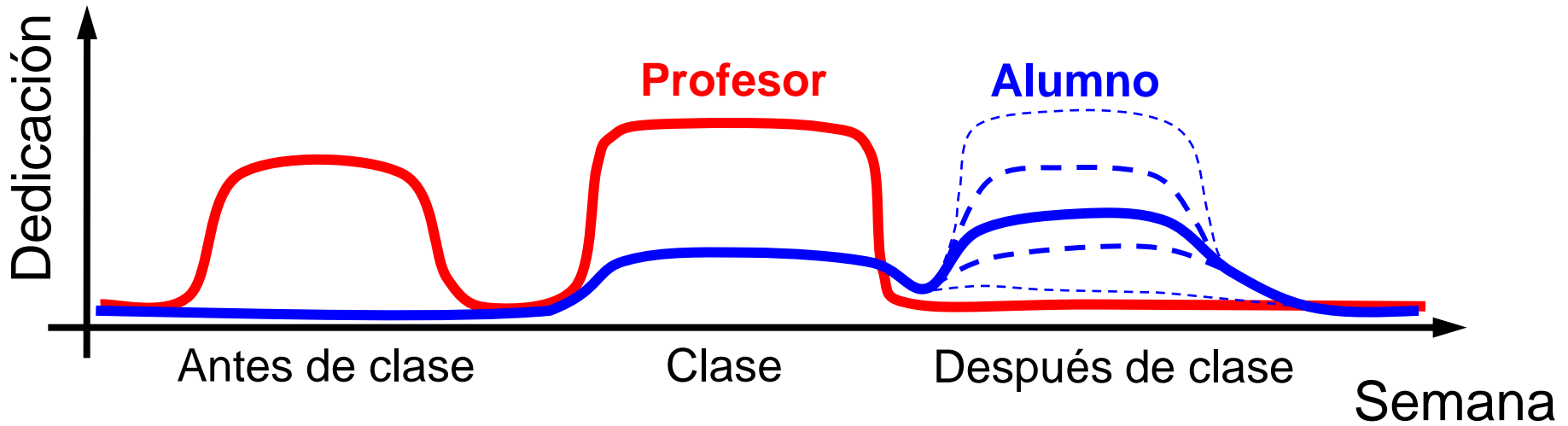
→Subgrupo 2: martes, 12:20 a 14:00, laboratorio 1.5

→Subgrupo 3: jueves, 10:40 a 12:20, laboratorio 2.3

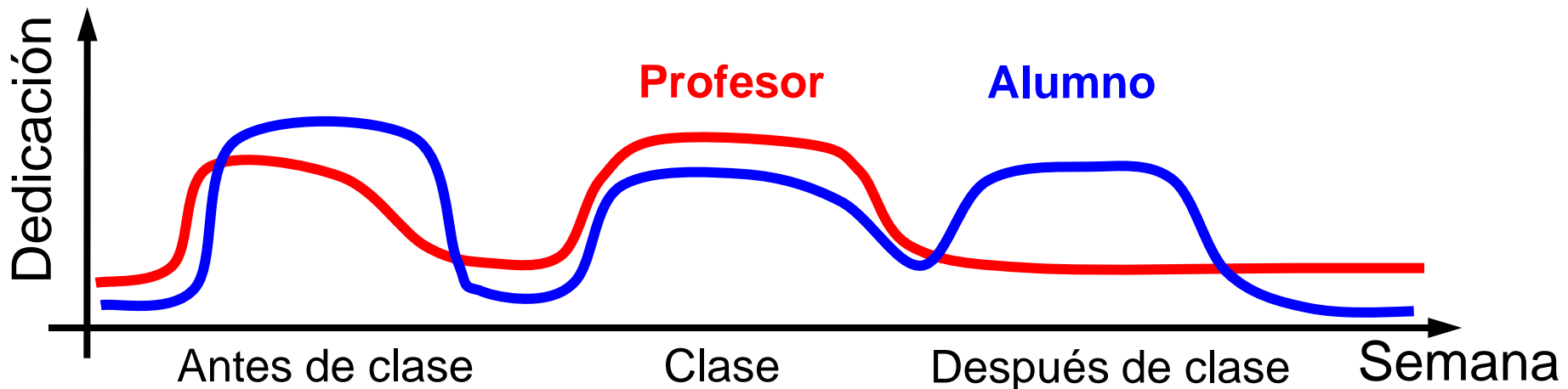
# El Problema con los Exámenes

## Espacio Europeo de Educación Superior (EEES)

- Modelo educativo anterior

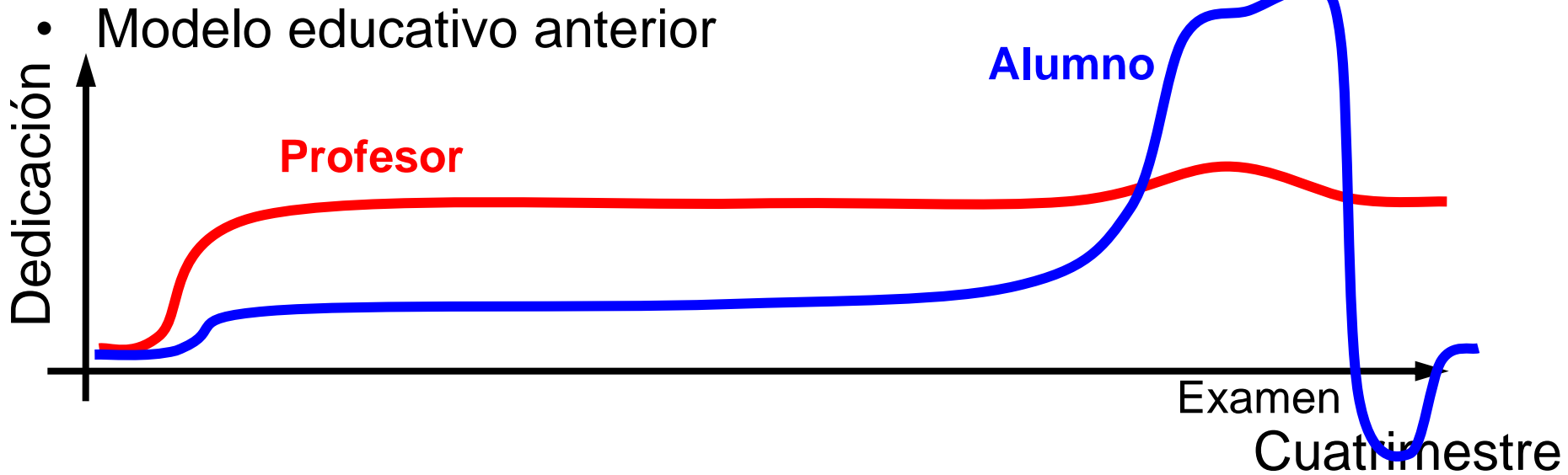


- Nuevo modelo educativo

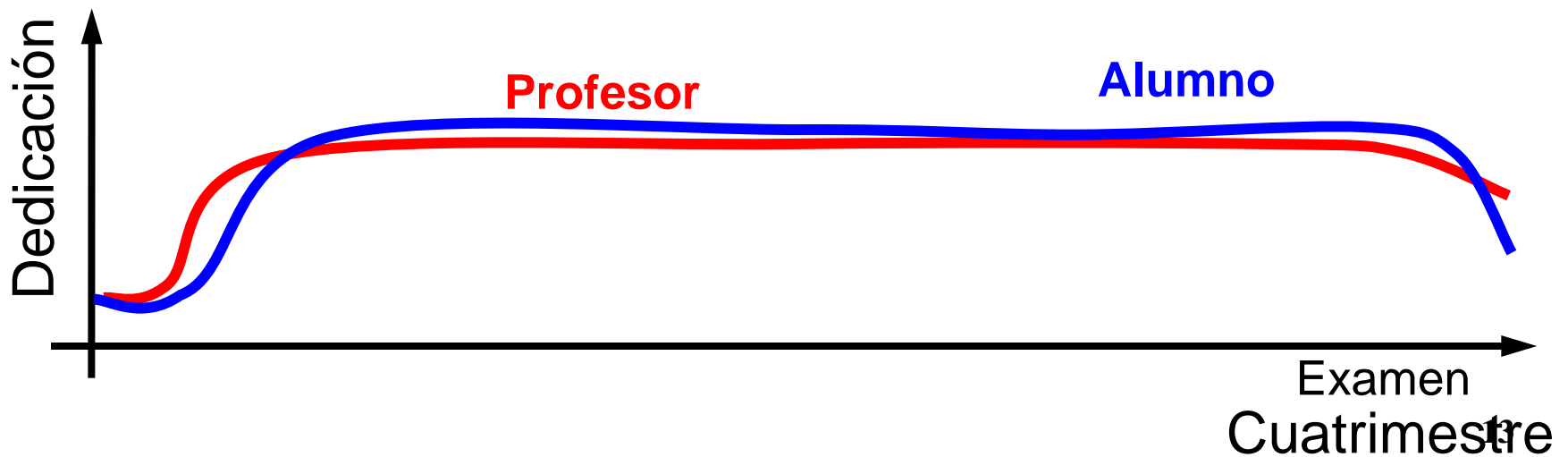


# El Problema con los Exámenes

## Espacio Europeo de Educación Superior (EEES)



- Nuevo modelo educativo



# Evaluación Continua

## Principios

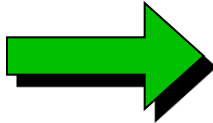
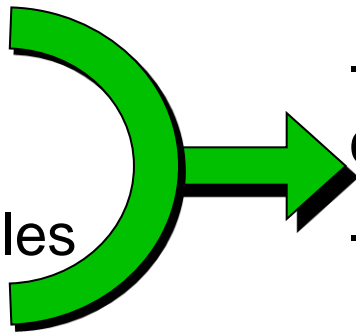
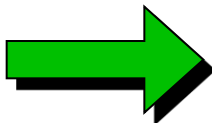
- Evaluación continua del trabajo a lo largo de todo el curso, no atracón de última hora.
- Para cada tema, se realizan determinadas actividades. Si se superan, el tema queda convalidado.
- Si alguien convalida solo algunos temas, puede recuperar los que queden en el examen, pero siempre con la asistencia a clase.
- La asistencia a clase es obligatoria (mínimo del 80%).

# Actividades de Evaluación Continua

- **Asistencia a clase:** se pasará lista de asistencia.
- **Resúmenes:** leer temas del texto guía, entregar resúmenes: una sola hoja escrita a mano.
- **Prácticas entregables:** prácticas de los temas 1 y 4 que eliminan materia para el examen.
- **Examen de tipo preguntas cortas (temas 2 y 3):** ejecutar algoritmos, relacionar cosas, aspectos esenciales.
- **Práctica temas 2 y 3:** implementación y manejo de estructuras de datos: lenguajes C/C++, sobre Linux.
- Y por supuesto... **¡¡El juez on-line!!**

# Actividades de Evaluación Continua

## Algoritmos y Estructuras de Datos I

1. Abstracciones y especificaciones  Ejercicios de Maude (grupos de 2) ~15-oct
2. Conjuntos y diccionarios
3. Repr. de conjuntos mediante árboles  -Examen preg. cortas ~18-nov  
-Práctica ~6-dic
4. Grafos  Ejercicios de programación (individual) ~27-dic



Asistencia a clase y entrega de resúmenes



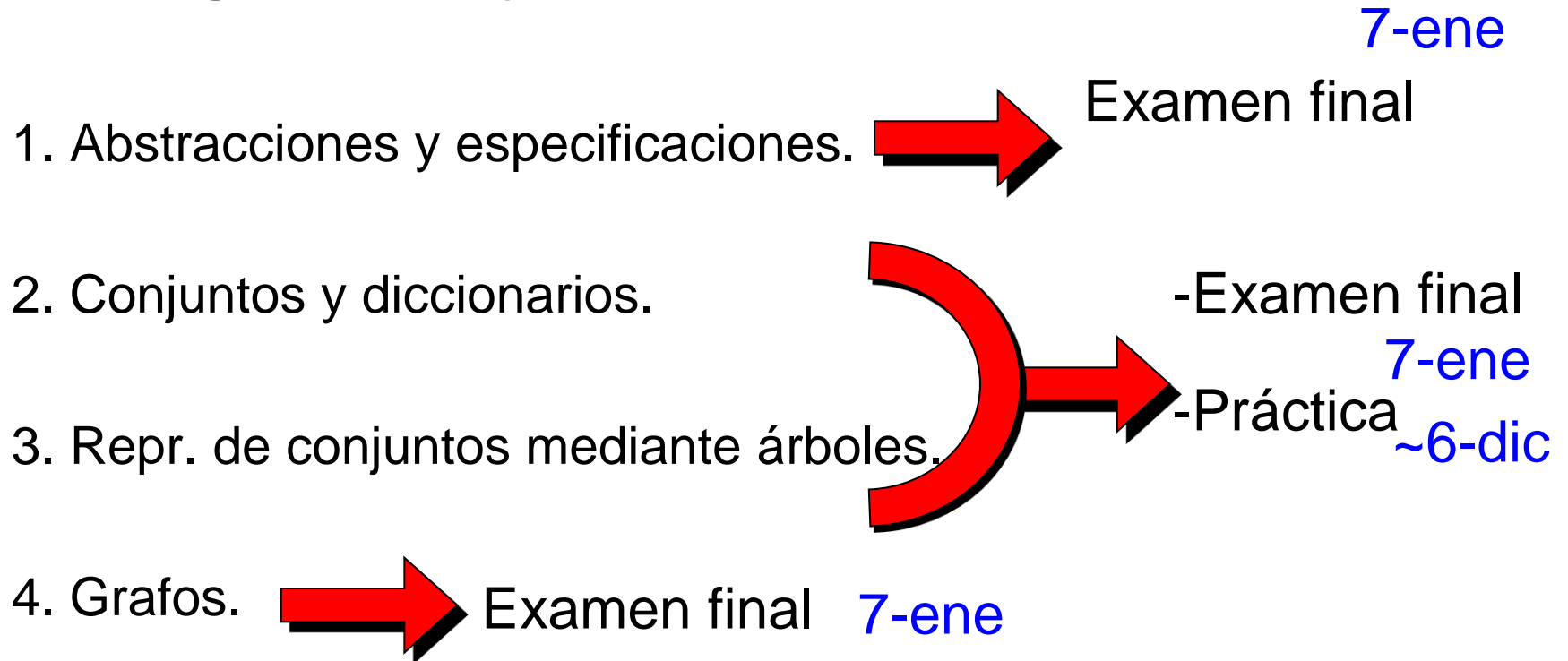
# Evaluación Alternativa

## Principios y Actividades

- Examen final. Mínimo una pregunta por tema.
- Práctica de los temas 2 y 3: implementación y manejo de estructuras de datos; lenguajes C/C++, sobre Linux
- No se requiere asistencia a clase ni otras actividades.

# Actividades de Evaluación Alternativa

## Algoritmos y Estructuras de Datos I



# Práctica temas 2 y 3

## *¿Práctica: Implementación y manejo de estructuras de datos.*

- *Ejercicios básicos.*
- *Implementación de tabla de dispersión.*
- *Diccionarios mediante árboles.*
- *Editor de texto. ?*

# Otras actividades...

- **Notas adicionales finales:**

*(hasta 1 punto sobre la nota final, siempre que esté aprobada la asignatura)*

- Participación en clase
- Ejercicios en C
- Concurso de programación ACM Contest

- **Comodines:**

*(valen por dos ejercicios de la actividad de especificaciones formales, o por dos ejercicios del examen de preguntas cortas, o por un ejercicio de la práctica de grafos)*

- Ejercicios en C
- Concurso de programación ACM Contest

# 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

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 Ver Preguntar

**Programa** Browse... Enviar Imprimir

**Listados**  Envíos  Clasificación  Preguntas  Impresiones

Actualizar cada 5 minutos con 15 líneas Ayuda Salir

**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, titled "AC: Extractor de Practicas", shows a file explorer with "AntiCopias v1.7 r211" selected. A "Bzip2\_ncd\_sim" window is open, displaying a network graph with nodes 1, 2, 3, 8, 10, and 28. Below the graph is a histogram showing a distribution of values. The bottom of the interface has a file list with "copias.s08.bt" and a "Buscar" button. Two code editors are open, showing C++ code for image processing.

```
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_cmp.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
```

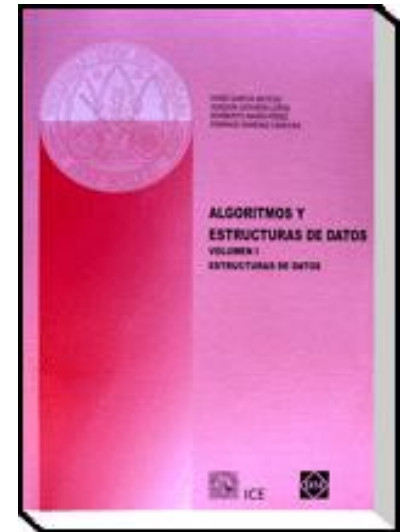
# Tutorías

- **Lunes, 11:30-13:30**
- **Martes , 11:15 a 12:15**
- **Tutorías virtuales**
- **Despacho 2.27 (2ª planta Fac. Informática)**
- **E-mail:** [nmarin@um.es](mailto:nmarin@um.es)
- **Web asignatura:** [webs.um.es/nmarin/](http://webs.um.es/nmarin/)

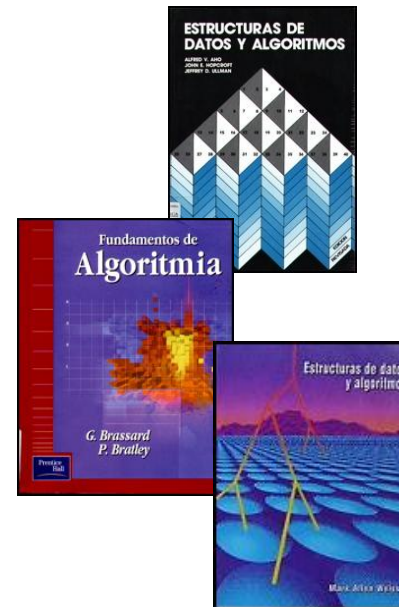
# Bibliografía

- **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



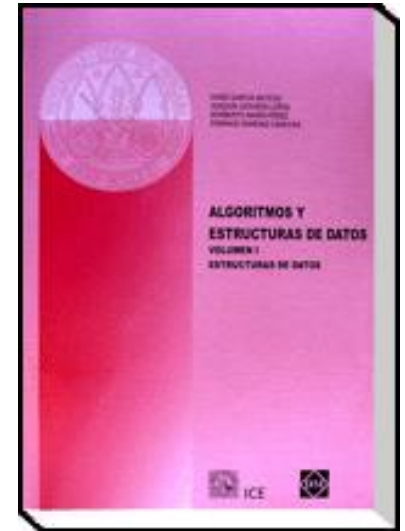
- **Estructuras de datos y algoritmos**  
A.C. Aho, J.E. Hopcroft, J.D. Ullman  
Addison-Wesley Iberoamericana, 1988
- **Fundamentos de Algoritmia**  
G. Brassard, P. Bratley  
Prentice-Hall, 1998
- **Estructuras de datos y algoritmos**  
Mark Allen Weiss  
Addison-Wesley Iberoamericana, 1995



(más en la web de la asignatura)

# Ejercicios para casa

- Leer las secciones 2.1 y 2.2 del texto guía.
- Preparar un resumen en un folio por las dos caras **ESCRITO A MANO**.
- Entregar la siguiente clase de Teoría, con el siguiente formato cabecera:



Nombre alumno, Grupo, AED Cap.1, Fecha, Horas estimadas