

Grupo ARCOS

**uc3m** | Universidad **Carlos III** de Madrid

# Estructura de Computadores

## Objetivos y presentación del curso

### Estructura de Computadores

Grado en Ingeniería Informática

Grado en Matemática aplicada y Computación

Doble Grado en Ingeniería Informática y Administración de Empresas



# Presentación



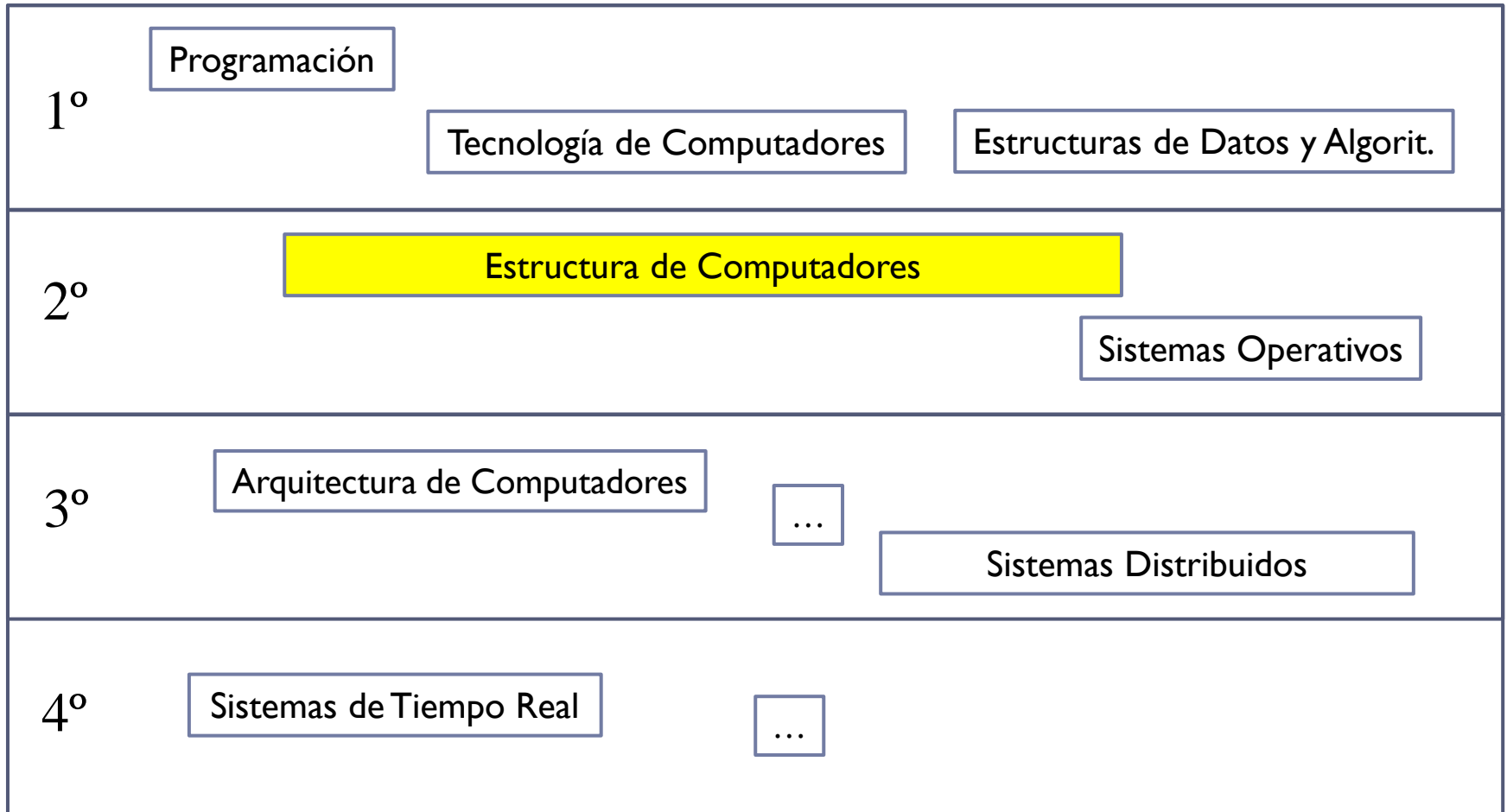
- ▶ **Información general**
- ▶ **Desarrollo del curso**
- ▶ **Sistema de evaluación**

# Estructura de Computadores en la UC3M

- ▶ Es una asignatura de segundo curso y primer cuatrimestre que se imparte en tres titulaciones:
  - ▶ [Grado en Ingeniería Informática](#)
  - ▶ [Grado en Matemática aplicada y Computación](#)
  - ▶ [Doble Grado en Ingeniería Informática y Administración de Empresas](#)

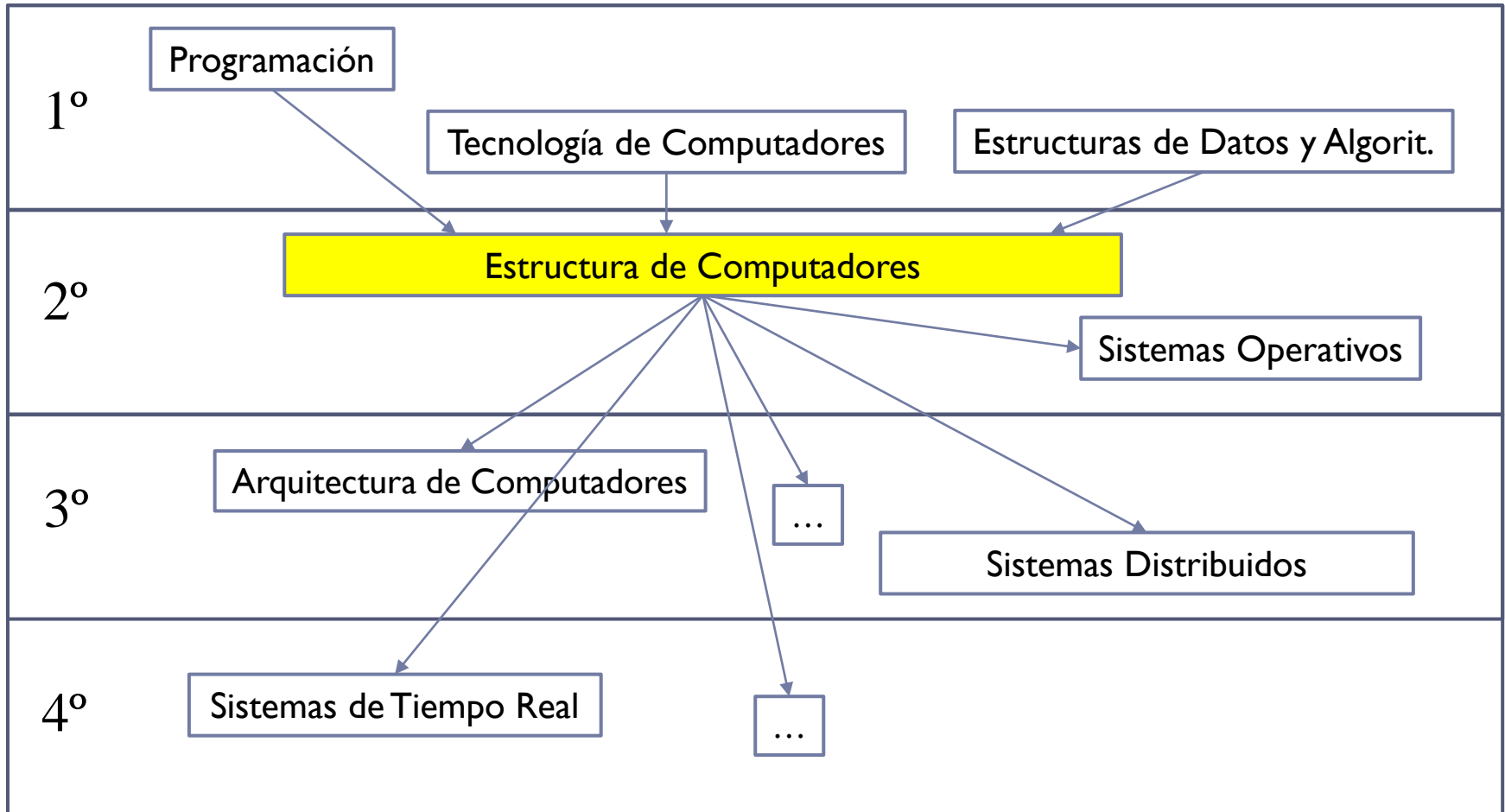
# Estructura de Computadores

## Grado en Ingeniería Informática



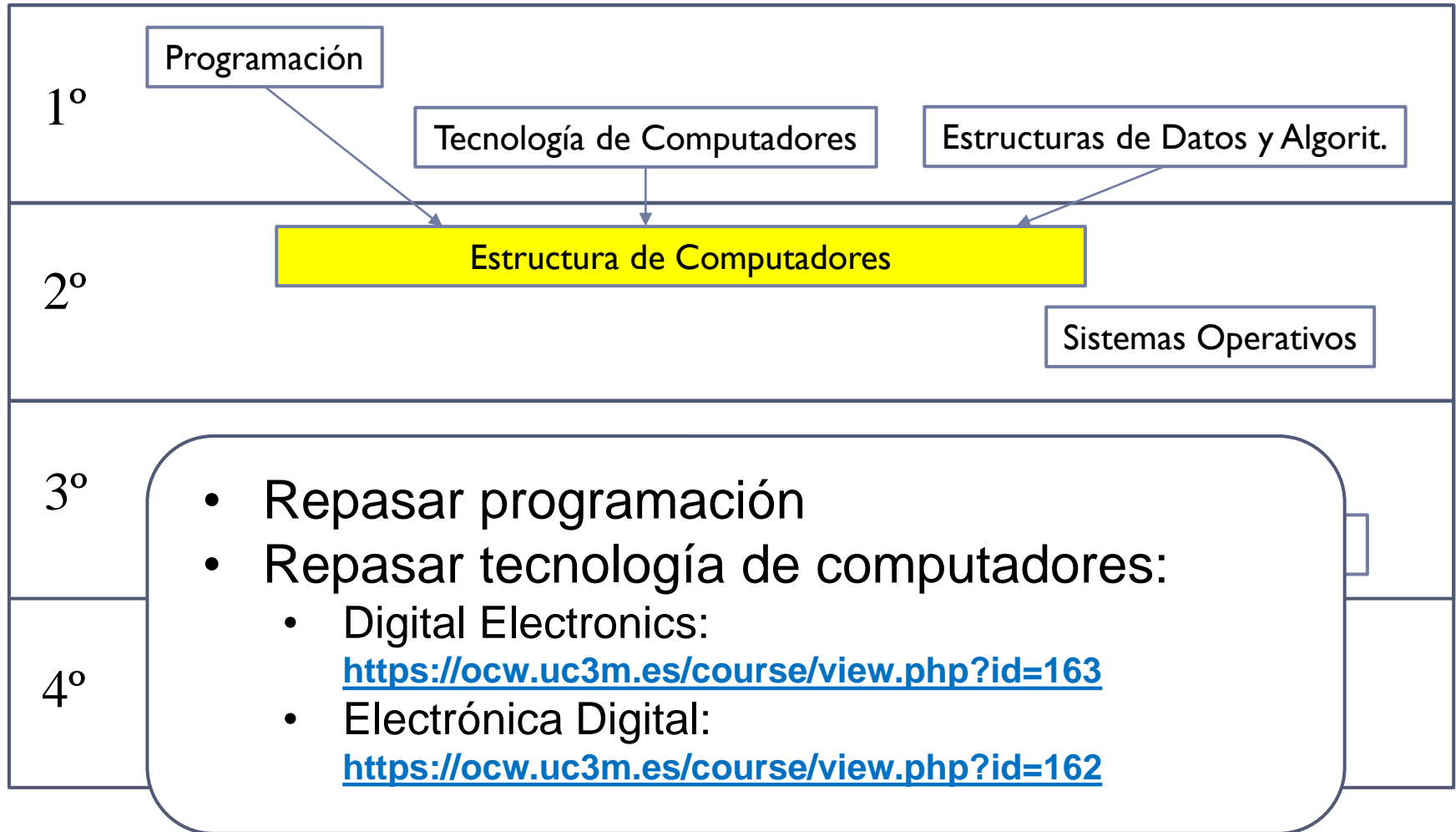
# Estructura de Computadores

## Grado en Ingeniería Informática



# Estructura de Computadores

## Grado en Ingeniería Informática



# Ficha resumida

- ▶ ESTRUCTURA DE COMPUTADORES
- ▶ **Coordinador**
- ▶ **Objetivos**
- ▶ **Programa**
- ▶ **Materiales: bibliografía**

# Ficha resumida

## Coordinador

- ▶ ESTRUCTURA DE COMPUTADORES
- ▶ OBLIGATORIA / FORMACIÓN BÁSICA
- ▶ CURSO: 2º
- ▶ CUATRIMESTRE: 1º
- ▶ CRÉDITOS ECTS: 6
- ▶ Profesor Coordinador: Félix García Carballeira  
(felix.garcia@uc3m.es)



# Ficha resumida

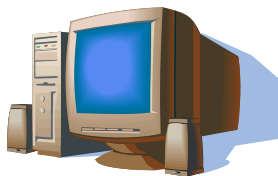
## Objetivos del curso

- ▶ ESTRUCTURA DE COMPUTADORES
- ▶ Conocer y entender  
los principales componentes y el funcionamiento básico  
de un computador

# Ficha resumida

## Objetivos del curso

- ▶ ESTRUCTURA DE COMPUTADORES
- ▶ Conocer y entender los principales componentes y el funcionamiento básico de un computador



# Ficha resumida

## Objetivos del curso

- ▶ ESTRUCTURA DE COMPUTADORES
- ▶ Conocer y entender los principales componentes y el funcionamiento básico de un computador

### Ejemplo 1

- ▶ Sea el siguiente código:

```
int n;  
n = 40000;  
printf("%d \n", n*n ); // 40000²  
  
n = 50000;  
printf("%d \n", n*n ); // 50000²
```

- ▶ Produce la siguiente salida:

```
1600000000  
-1794967296
```

- ▶ ¿Es correcto?, ¿Cuál es el problema?

### Ejemplo 2

- ▶ Sea el siguiente código:

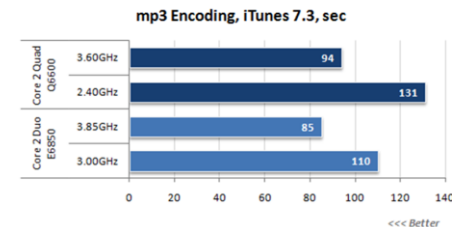
```
float x, y, z;  
  
x = 1.0e20; y = -1.0e20; z = 3.14;  
  
printf("%f\n", (x + y) + z);  
printf("%f\n", x + (y + z));
```

- ▶ Produce la siguiente salida:

```
3.140000  
0.000000
```

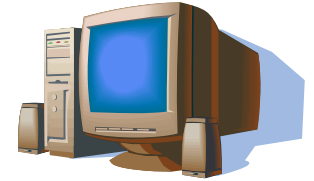
- ▶ ¿Se cumple  $(x+y) + z == x + (y+z)$ ?

### Ejemplo 3



- ▶ ¿Es más rápido un procesador con dos núcleos o un procesador con cuatro núcleos?

# Ejemplo 1



- ▶ Sea el siguiente código:

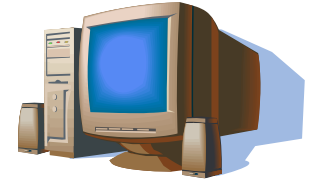
```
int n;  
n = 40000;  
printf("%d \n", n*n ); // 400002  
  
n = 50000;  
printf("%d \n", n*n ); // 500002
```

- ▶ Produce la siguiente salida:

```
1600000000  
-1794967296
```

- ▶ ¿Es correcto?, ¿Cuál es el problema?

# Ejemplo 2



- Sea el siguiente código:

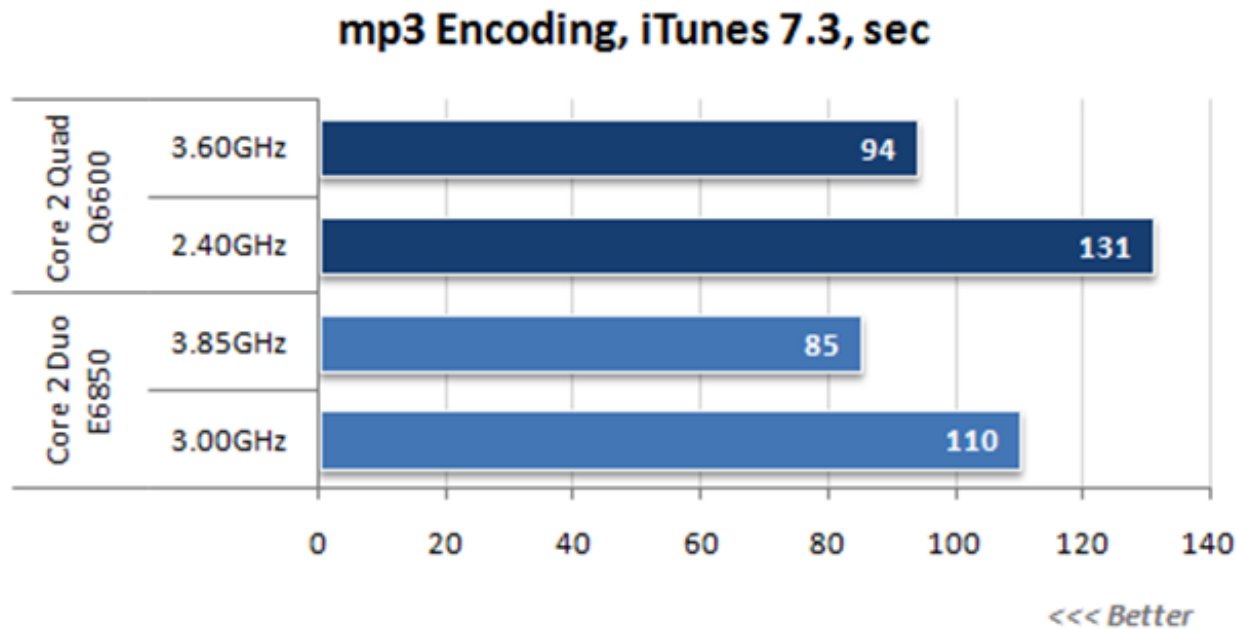
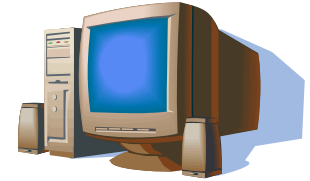
```
float x, y, z;  
  
x = 1.0e20; y = -1.0e20; z = 3.14;  
  
printf("%f\n", (x + y) + z);  
printf("%f\n", x + (y + z));
```

- Produce la siguiente salida:

```
3.140000  
0.000000
```

- ¿Se cumple  $(x+y) + z == x + (y+z)$  ?

# Ejemplo 3



- ¿Es más rápido un procesador con dos núcleos o un procesador con cuatro núcleos?

# Ficha resumida

## Objetivos del curso

- ▶ ESTRUCTURA DE COMPUTADORES
- ▶ Conocer y entender los principales componentes y el funcionamiento básico de un computador

### Ejemplo 1



- ▶ Sea el siguiente código:

```
int n;  
n = 40000;  
printf("%d \n", n*n ); // 40000²  
  
n = 50000;  
printf("%d \n", n*n ); // 50000²
```

- ▶ Produce la siguiente salida:

```
1600000000  
-1794967296
```

- ▶ ¿Es correcto?, ¿Cuál es el problema?

### Ejemplo 2



- ▶ Sea el siguiente código:

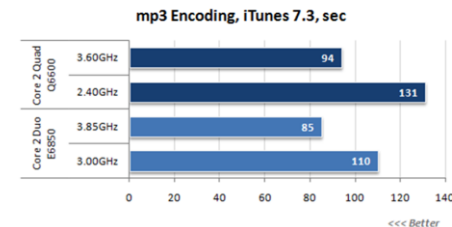
```
float x, y, z;  
  
x = 1.0e20; y = -1.0e20; z = 3.14;  
  
printf("%f\n", (x + y) + z);  
printf("%f\n", x + (y + z));
```

- ▶ Produce la siguiente salida:

```
3.140000  
0.000000
```

- ▶ ¿Se cumple  $(x+y) + z == x + (y+z)$ ?

### Ejemplo 3



- ▶ ¿Es más rápido un procesador con dos núcleos o un procesador con cuatro núcleos?

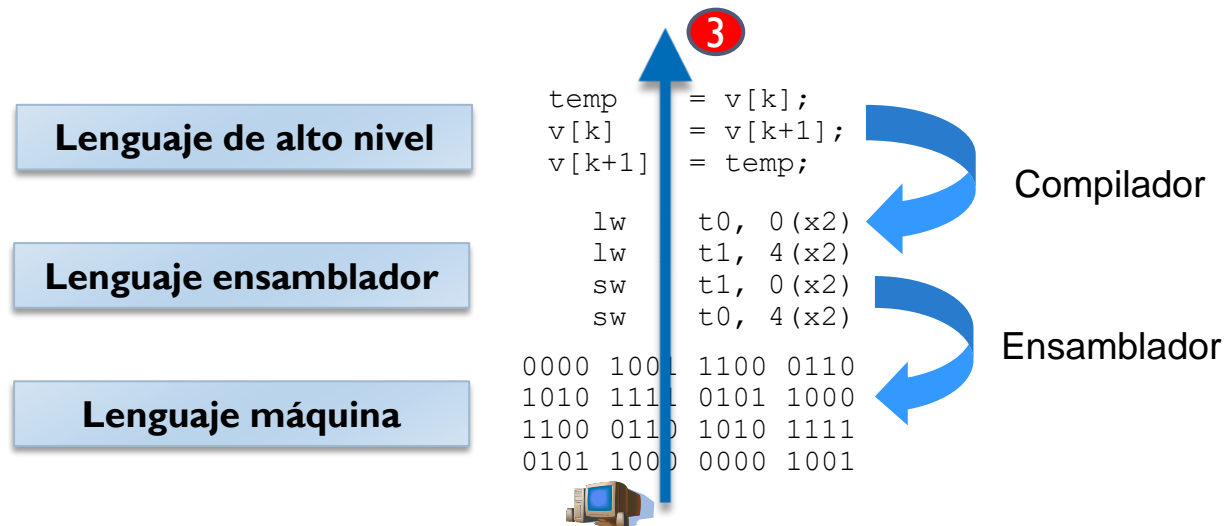
# Ficha resumida

## Objetivos del curso

### ► ESTRUCTURA DE COMPUTADORES

### ► Conocer y entender

los principales componentes y el funcionamiento básico de un computador





# Ficha resumida

## Programa

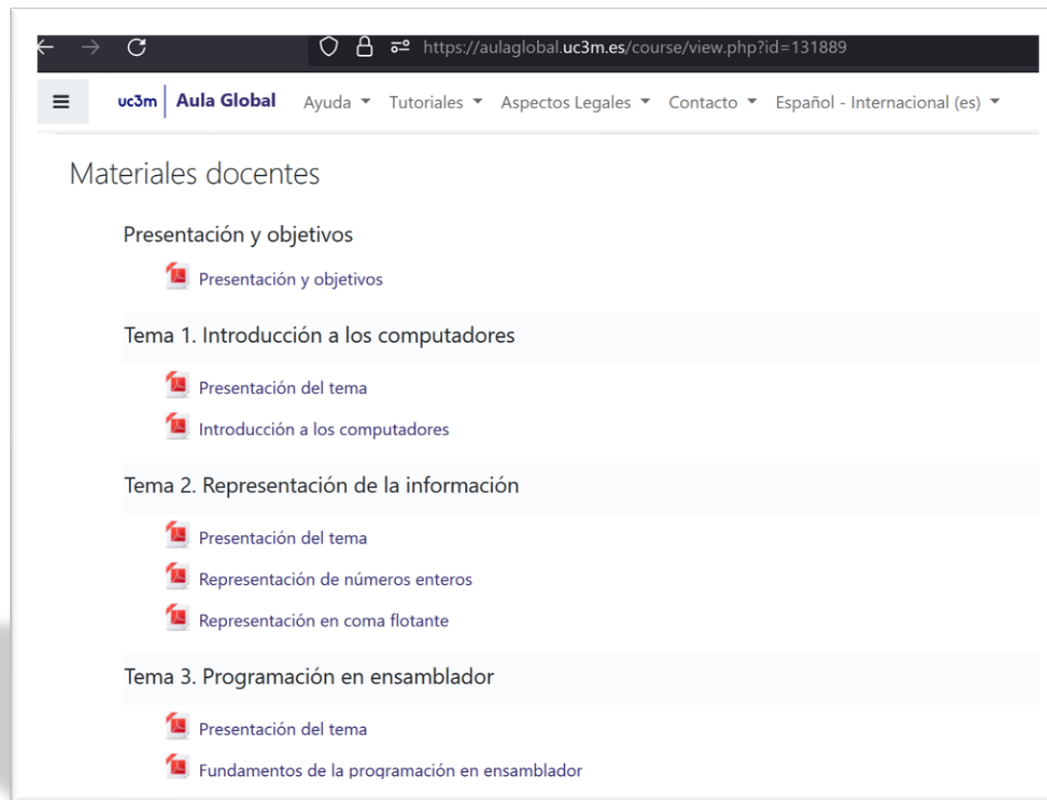


- ▶ Tema 1. Introducción a los computadores
- ▶ Tema 2. Representación de la información
- ▶ Tema 3. Fundamentos de la programación en ensamblador
- ▶ Tema 4. El procesador
- ▶ Tema 5. Sistemas de memoria
- ▶ Tema 6. Sistemas de entrada/salida

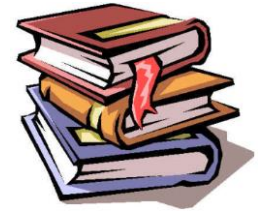
# Ficha resumida

## Contenido de la asignatura

### ► En Aula Global



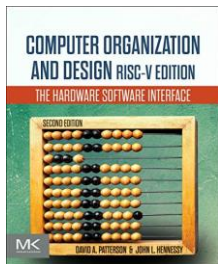
# Bibliografía básica



- ▶ **Problemas resueltos de Estructura de Computadores**  
F. García Carballera, J. Carretero Pérez,  
J. D. García, D. Expósito,  
Segunda edición,  
Editorial Paraninfo, 2015

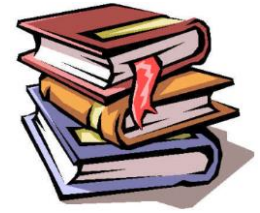


- ▶ **Computer Organization and Design  
The Hardware/Software Interface**  
D.A. Patterson, J. Hennessy  
Quinta edición, 2014

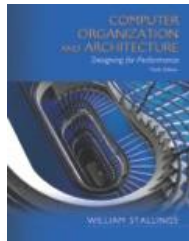


- ▶ **Computer Organization and Design RISC-V Edition:  
The Hardware Software Interface,**  
David A. Patterson, J. L. Hennessy,  
Segunda edición, 2021

# Bibliografía complementaria



- **Fundamentos de Sistemas Digitales.**  
Thomas L. Floyd  
Editorial Pearson, 2016



- **Computer Organization and Architecture.**  
William Stallings  
Décima edición,  
Editorial Pearson, 2016

# Materiales complementarios

- ▶ [Computer History Museum](#)
- ▶ [Museo virtual de la Informática,  
Universidad de Castilla-la Mancha](#)
- ▶ <https://www.computer.org/cms/Computer.org/Publications/timeline.pdf>
- ▶ [The EDSAC Simulator](#)
- ▶ [IBM archives](#)
- ▶ [Charles Babbage Institute](#)
- ▶ [Museo histórico de la Informática,  
Universidad Politécnica de Madrid](#)

# Presentación



- ▶ **Información general**
- ▶ **Desarrollo del curso**
- ▶ **Sistema de evaluación**

# Desarrollo del curso



- ▶ 14 semanas en total (todas las sesiones presenciales)
  - ▶ 14 sesión de 100 min. en grupo magistral
  - ▶ 15 sesión de 100 min. en grupo reducido
    - ▶ 4 de estas sesiones son de laboratorio (presencial)
    - ▶ 11 sesiones presenciales

# Desarrollo del curso



- ▶ 14 semanas en total (todas las sesiones presenciales)
  - ▶ 14 sesión de 100 min. en grupo magistral
  - ▶ 15 sesión de 100 min. en grupo reducido
    - ▶ 4 de estas sesiones son de laboratorio (presencial)
    - ▶ 11 sesiones presenciales

100 m	100 m	4,6 h (trabajo personal)
-------	-------	--------------------------



dedicación media semanal (8 horas)



# Profesores y aulas

## Leganés, Grado en Ing. Informática



Estructura de Computadores (cod. 13874), 6 ECTS

→ Este es un horario general de la asignatura. Aquí puedes ver el horario completo.

→ La información de los horarios de esta titulación ha sido generada de forma automática. Los cambios realizados durante esta jornada no se reflejarán instantáneamente.

→ La equivalencia de semanas para los horarios puede consultarla en la siguiente tabla (la fecha corresponde al lunes de la semana en cuestión).

1er Cuatrimestre	S1→ 22/08	S2→ 29/08	S3→ 05/09	S4→ 12/09	S5→ 19/09	S6→ 26/09	S7→ 03/10	S8→ 10/10	S9→ 17/10	S10→ 24/10	S11→ 31/10	S12→ 07/11	S13→ 14/11	S14→ 21/11	S15→ 28/11	S16→ 05/12	S17→ 12/12	S18→ 19/12	S19→ 26/12	S20→ 02/01	S21→ 09/01	S22→ 16/01	S23→ 23/01
------------------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------	------------

### Grupo 81

Responsable: GARCIA CARBALLEIRA, FELIX

Responsable grupo agregado: CALDERON MATEOS, ALEJANDRO

Mar	11:00-13:00	Semanas: 3-17	Aulas: 2.3.D02
Vie	11:00-13:00	Semanas: 3-4, 6-7, 9, 11-12, 14-17	Aulas: 1.0.F01
Vie	11:00-13:00	Semanas: 5, 8, 10, 13	Aulas: INF 1.2.G.01

### Grupo 83

Responsable: CASARES ANDRES, MARIA GREGORIA

Responsable grupo agregado: CASARES ANDRES, MARIA GREGORIA

Mar	17:00-19:00	Semanas: 3-17	Aulas: 2.3.C04
Jue	17:00-19:00	Semanas: 3-4, 6-7, 9, 11-12, 14-17	Aulas: 1.0.C01
Jue	17:00-19:00	Semanas: 5, 8, 10, 13	Aulas: INF 7.0.J03 DUAL + TEL

### Grupo 84

Responsable: No especificado

Responsable grupo agregado: CASARES ANDRES, MARIA GREGORIA

Mar	17:00-19:00	Semanas: 3-17	Aulas: 2.3.C04
Mie	15:00-17:00	Semanas: 3-17	Aulas: 2.3.B01

### Grupo 87

Responsable: RINCON FUENTES, FRANCISCO DANIEL

Responsable grupo agregado: HERNANDEZ BRAVO, ANGEL

Mar	15:00-17:00	Semanas: 3-17	Aulas: 7.0.J06
Jue	19:00-21:00	Semanas: 3-4, 6-7, 9, 11-12, 14-17	Aulas: 7.1.H01
Jue	19:00-21:00	Semanas: 5, 8, 10, 13	Aulas: INF 7.0.J02 DUAL+ TEL

### Grupo 88

Responsable: PEREZ TRAPERO, ANTONIO

Responsable grupo agregado: HERNANDEZ BRAVO, ANGEL

Mar	15:00-17:00	Semanas: 3-17	Aulas: 7.0.J06
Vie	15:00-17:00	Semanas: 3-4, 6-7, 9, 11-12, 14-17	Aulas: 1.0.B03
Vie	15:00-17:00	Semanas: 5, 8, 10, 13	Aulas: INF 1.2.G.03 DUAL

### Grupo 89

Responsable: TESSIER FERNANDEZ, CARLOS

Responsable grupo agregado: HERNANDEZ BRAVO, ANGEL

Mar	15:00-17:00	Semanas: 3-17	Aulas: 7.0.J06
Jue	15:00-17:00	Semanas: 3-4, 6-7, 9, 11-12, 14-17	Aulas: 7.0.J01
Jue	15:00-17:00	Semanas: 5, 8, 10, 13	Aulas: INF 1.2.G.01

# Presentación

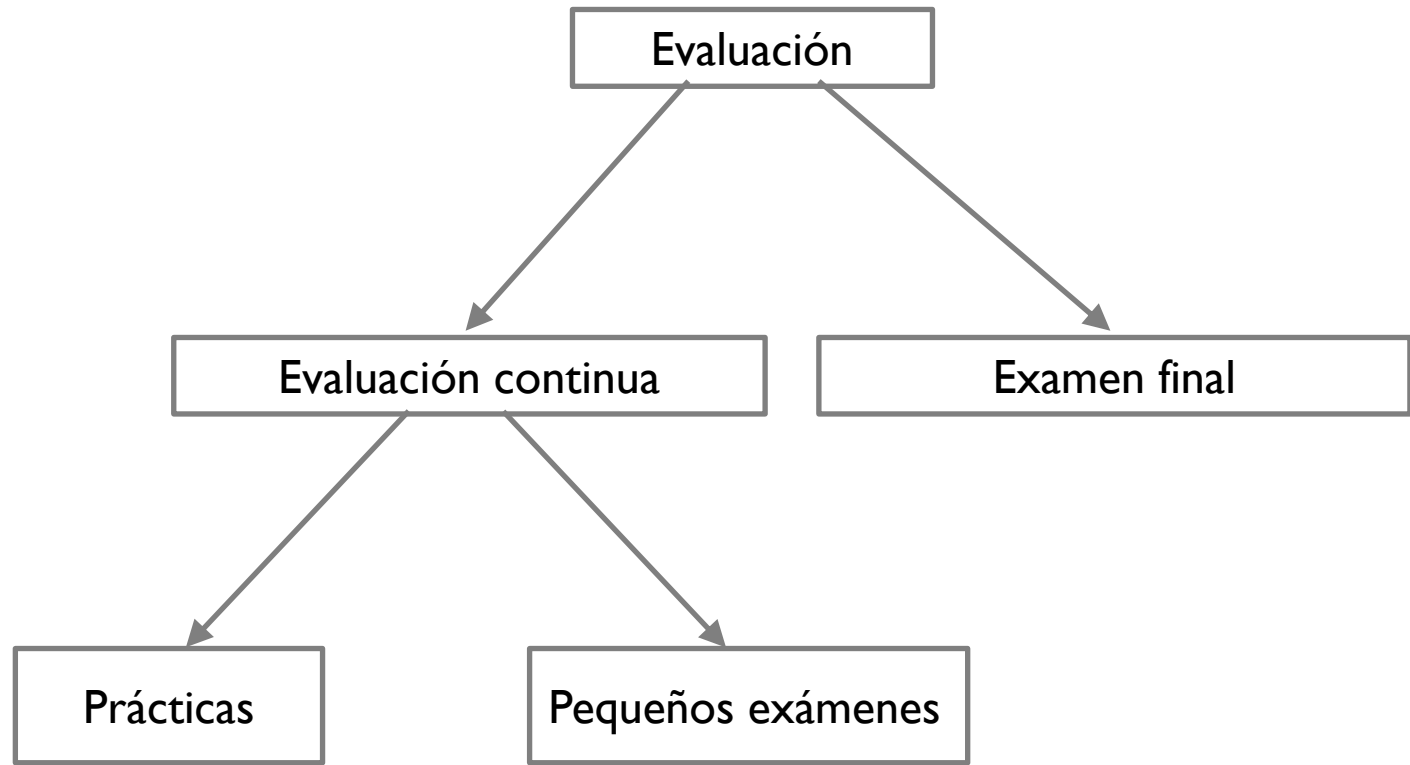


- ▶ **Información general**
- ▶ **Desarrollo del curso**
- ▶ **Sistema de evaluación**

# Evaluación del estudiante



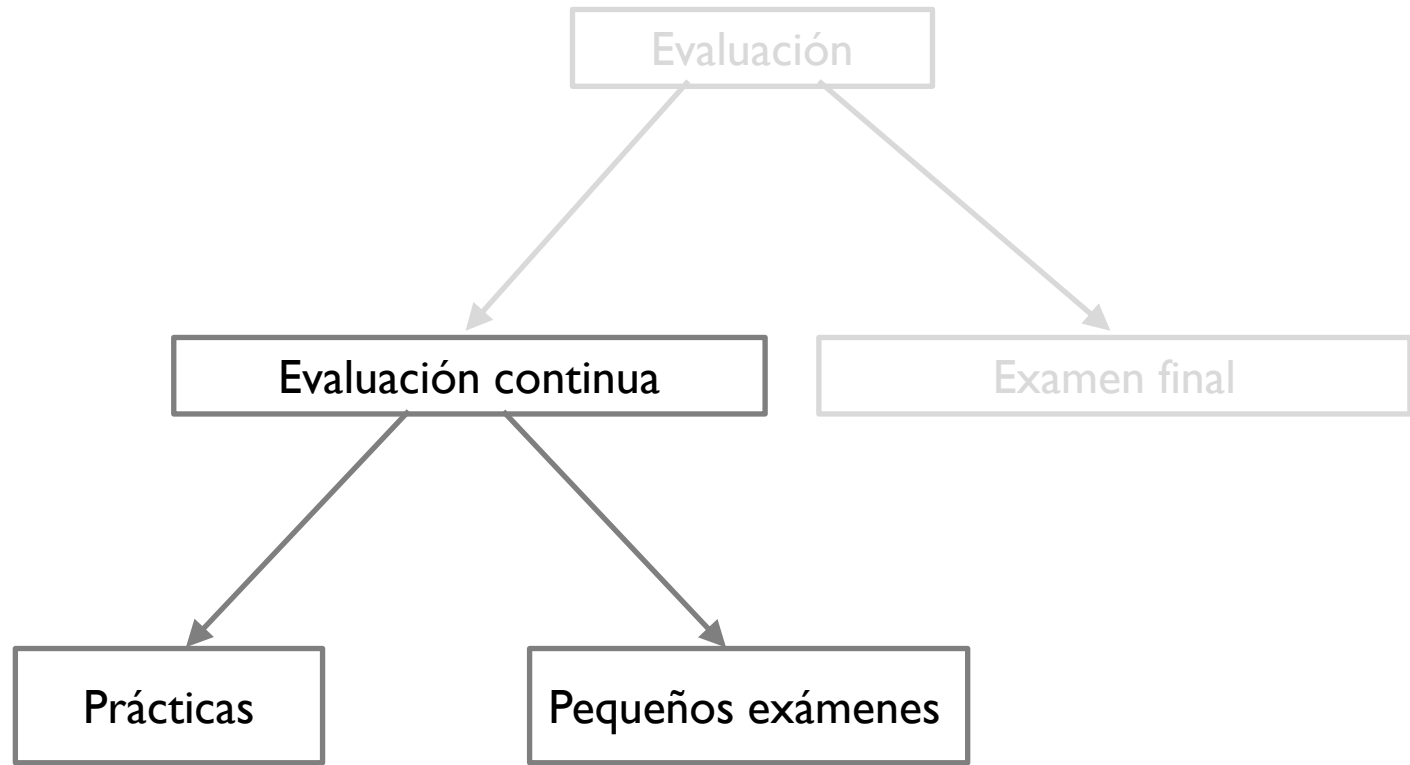
- La evaluación del estudiante se basará en:



# Evaluación del estudiante



- La evaluación del estudiante se basará en:



# Evaluación continua

## Prácticas

- ▶ Se realizarán **DOS** prácticas obligatorias
  - ▶ Nota mínima de cada práctica: **2**
  - ▶ Nota mínima media de todas las prácticas: **4**
- ▶ Pesos de cada práctica:
  - ▶ Práctica 1: 15%
  - ▶ Práctica 2: 15%
- ▶ Se realizarán en grupos de **dos** estudiantes
- ▶ En caso de que se detecte copia de prácticas, a ambas partes implicadas (copiados y copiadores) se les calificará con un 0

# Evaluación continua

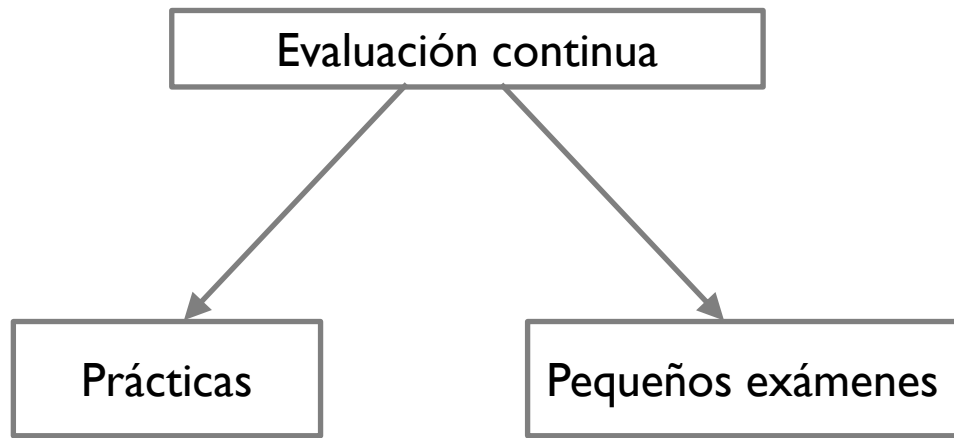
## Pequeños exámenes

- ▶ Se realizarán **TRES exámenes pequeños**
  - ▶ Duración: ~15 a ~20 minutos
  - ▶ Se evaluarán **todos** los conocimientos adquiridos por el alumno hasta ese momento
- ▶ Se realizan de forma individual.
- ▶ No se repetirá ningún examen.
- ▶ Salvo causa médica justificada con suficiente antelación, no se admitirá que un alumno realice el examen en un grupo distinto al que está matriculado.

# Evaluación continua



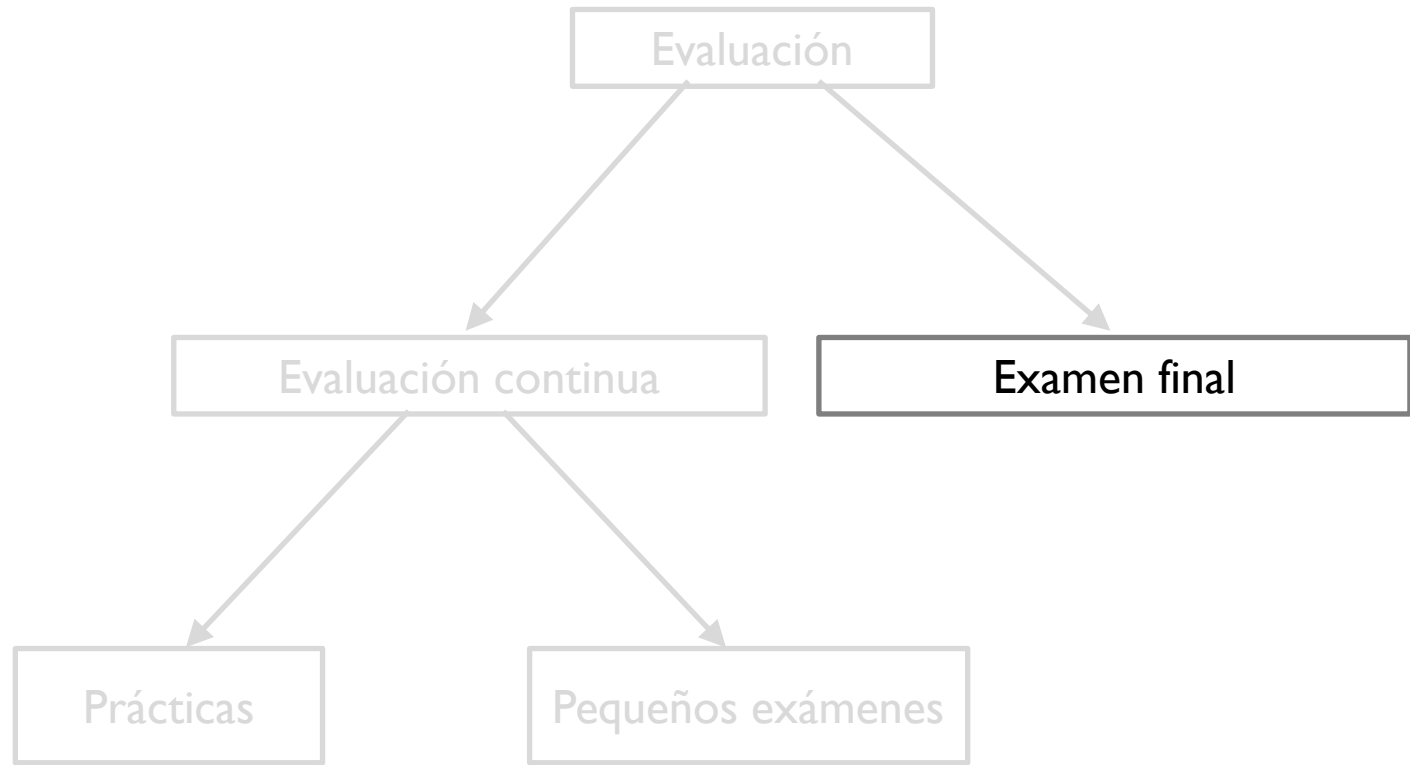
- ▶ Se **sigue la evaluación continua** cuando:
  - ▶ Se entrega **todas** las prácticas con:
    - ▶ Nota mínima de cada práctica: **2**
    - ▶ Nota mínima de todas las prácticas: **4**



# Evaluación del estudiante



- La evaluación del estudiante se basará en:





# Evaluación del estudiante:

## examen final



- ▶ Entra **todo** el contenido de la asignatura:  
todo el contenido teórico y práctico de la asignatura
  - ▶ La nota mínima en el examen final será de **4**
  - ▶ Si no se presenta a este examen aparecerá como **no presentado** (aunque se encuentre aprobado por evaluación continua).
- ▶ Para la realización del examen **no** se puede utilizar material de consulta alguno, tampoco se permite copiar.
- ▶ Será necesario presentar el **DNI** o **carnet universitario** para realizar la entrega del examen.

# Evaluación del estudiante



## 1. Convocatoria ordinaria

- ▶ Si se sigue la evaluación continua
- ▶ No se sigue la evaluación continua

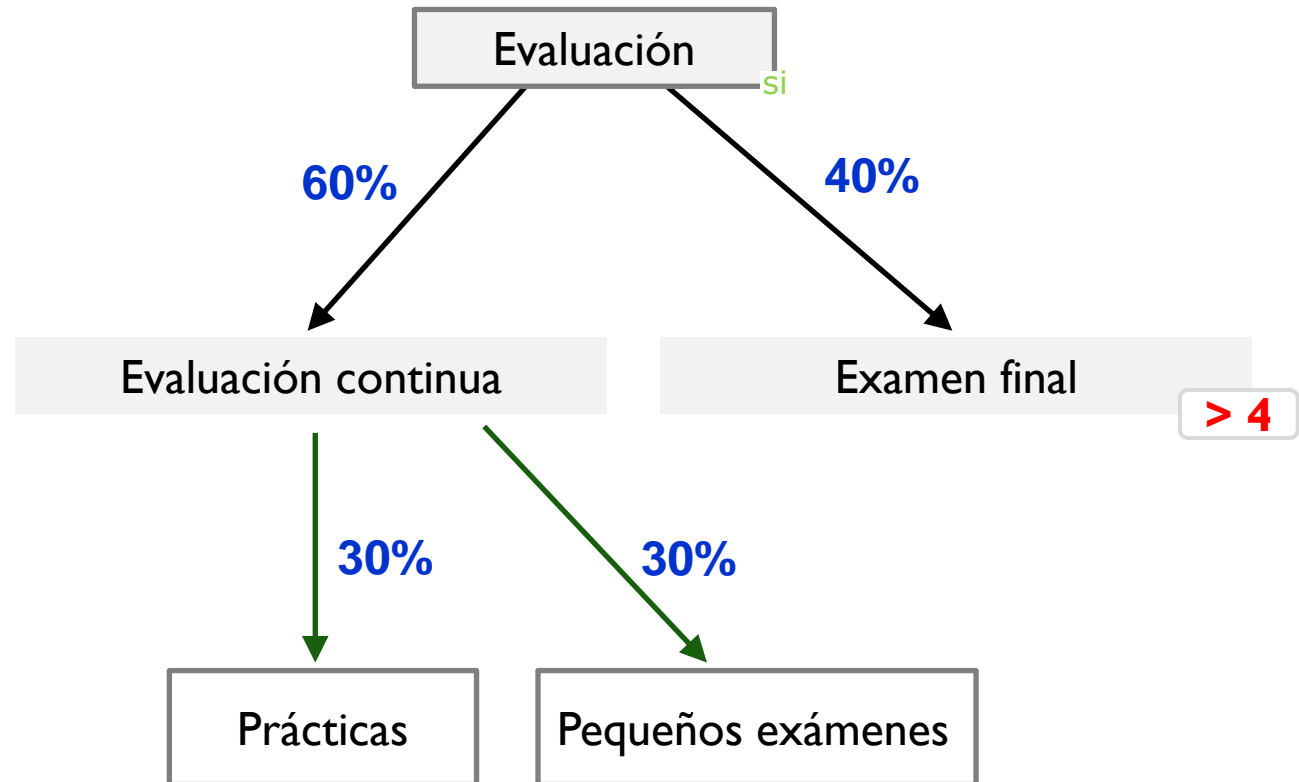
## 2. Convocatoria extraordinaria

- ▶ No se ha seguido la evaluación continua
- ▶ Si se ha seguido la evaluación continua

# Evaluación del estudiante



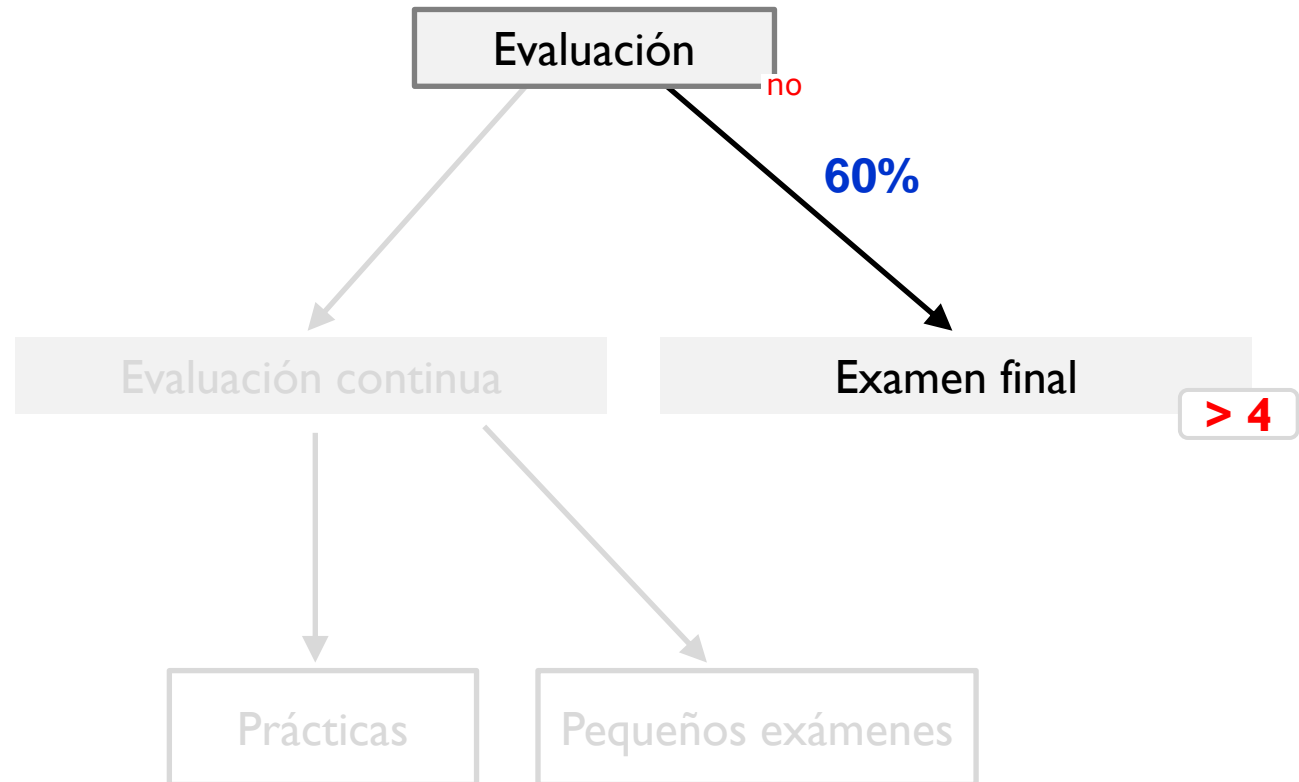
- Convocatoria ordinaria + se sigue continua:



# Evaluación del estudiante



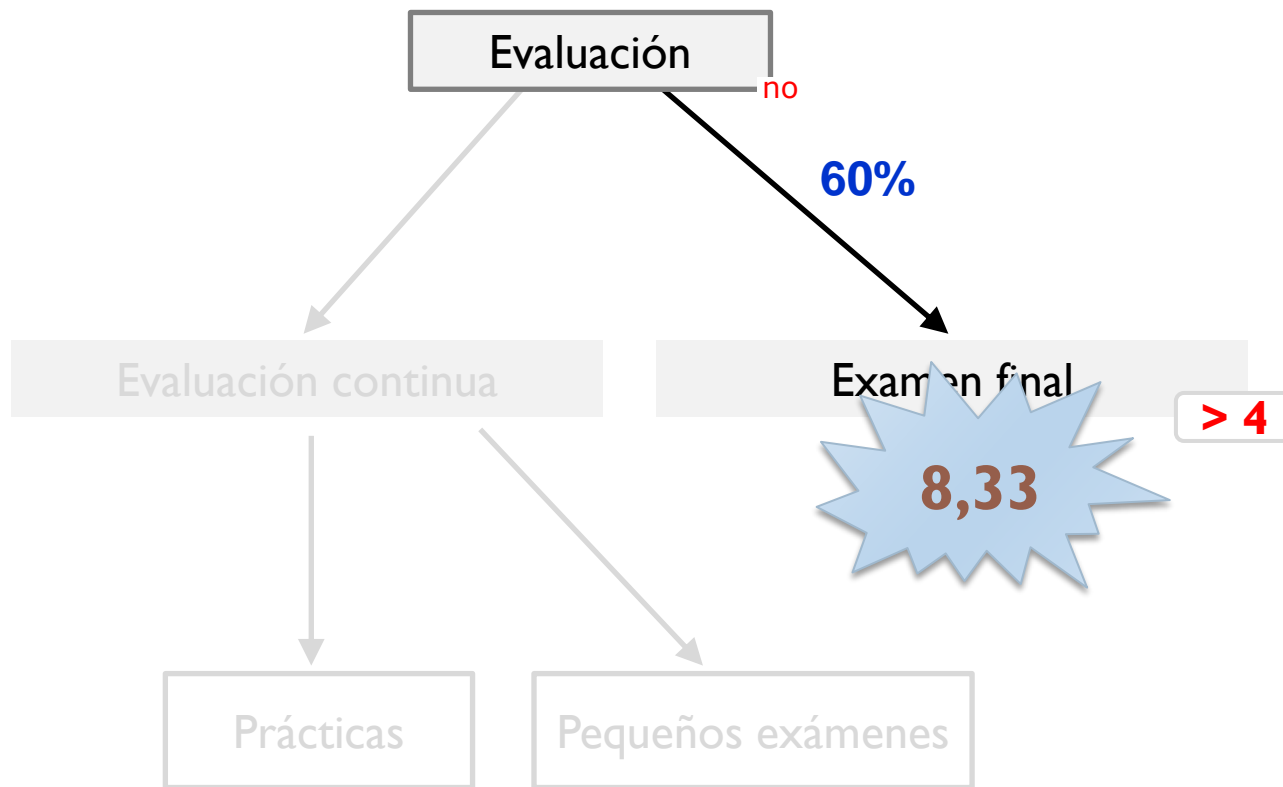
- Convocatoria ordinaria + no se sigue continua:



# Evaluación del estudiante



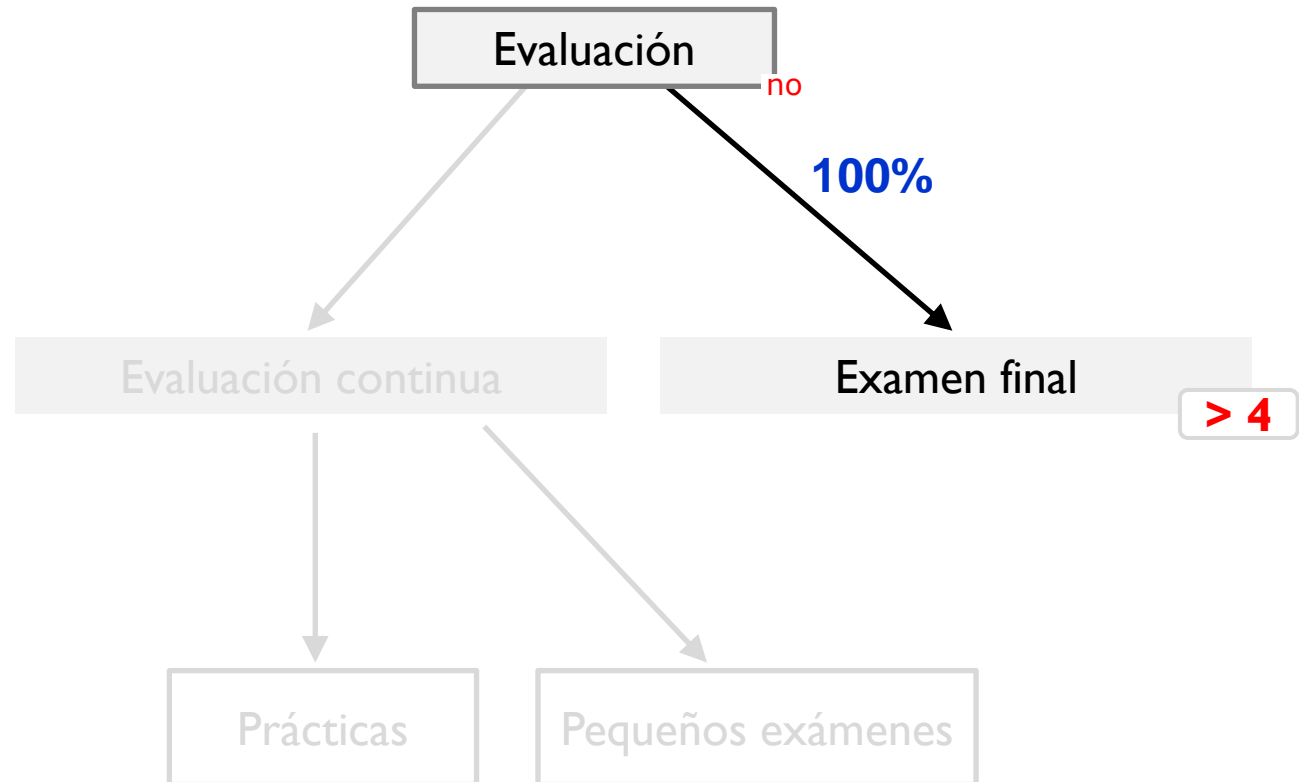
- Convocatoria ordinaria + no se sigue continua:



# Evaluación del estudiante



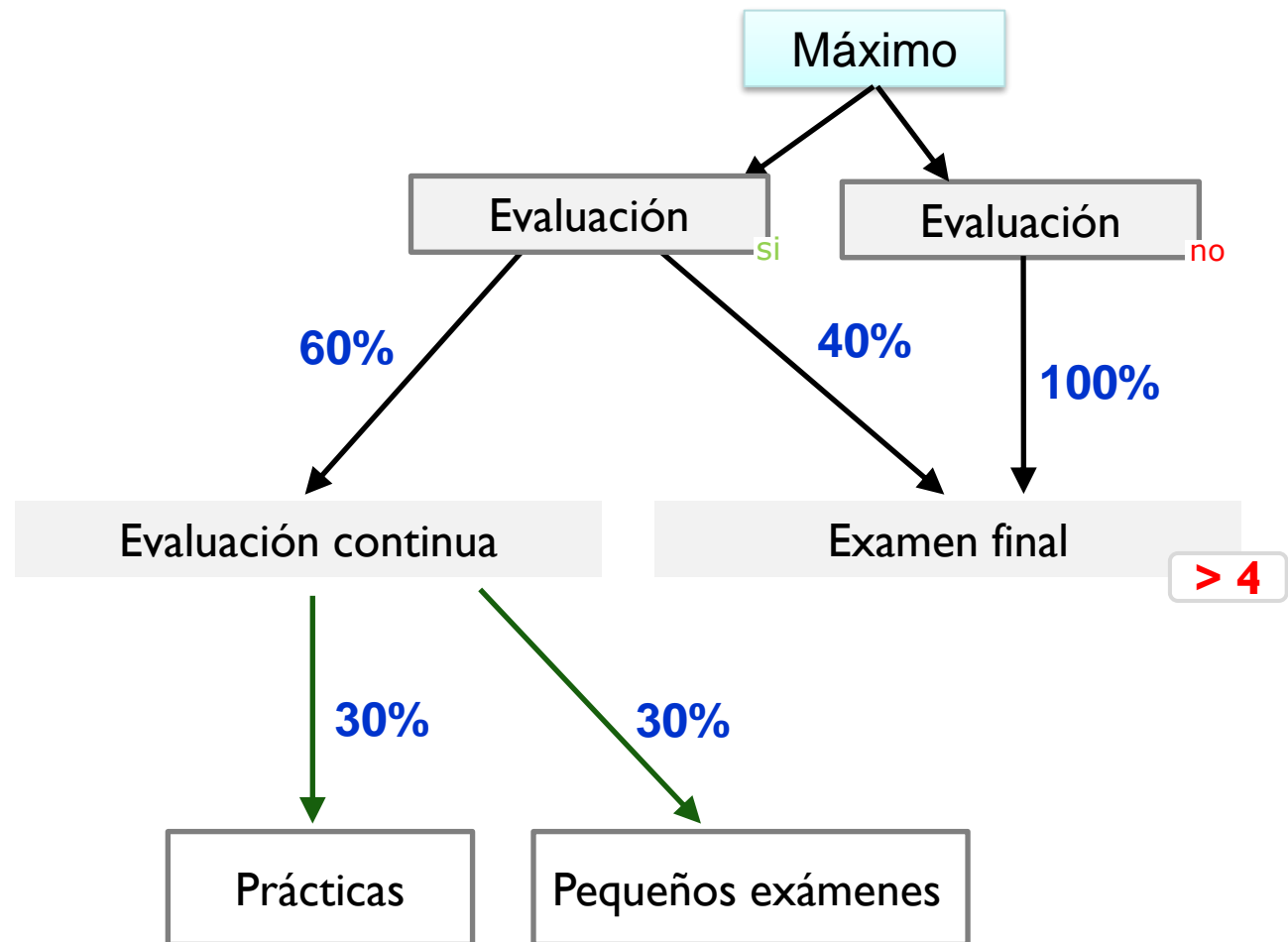
- Convocatoria **extra**ordinaria + no se sigue continua:



# Evaluación del estudiante



- Convocatoria **extra**ordinaria + se sigue continua:



# Importancia de la evaluación continua

	2014-2015	2015-2016	2016-2017	2017-2018	2018-2019	2019-2020
Estudiantes que <b>siguen la evaluación continua</b>	78%	74%	74%	71%	78%	88%
Estudiantes que <b>aprueban la evaluación continua</b>	74%	64%	67%	66%	76%	84%
Estudiantes que aprueban la evaluación continua respecto de los que la siguen	87%	87%	90%	89%	89%	95%
Estudiantes que <b>aprobaron</b> al final la <b>asignatura</b> aunque <b>abandonaron la evaluación continua</b>	< 1 %	< 1 %	< 1 %	< 1 %	<1%	<1%
Estudiantes que <b>aprobaron la evaluación continua y han aprobado la asignatura</b> al final	<b>92%</b>	<b>89%</b>	<b>85%</b>	<b>86%</b>	<b>94%</b>	<b>90%</b>
Estudiantes que completaron y suspendieron la evaluación continua han aprobado la asignatura al final	6%	7%	4%	6%	2%	<1%
<b>Estudiantes aprobados</b>	<b>67%</b>	<b>66%</b>	<b>65%</b>	<b>62%</b>	<b>78%</b>	<b>82%</b>
<b>Estudiantes no presentados</b>	<b>23%</b>	<b>20%</b>	<b>22%</b>	<b>26%</b>	<b>15%</b>	<b>9%</b>
<b>Estudiantes suspensos</b>	<b>10%</b>	<b>14%</b>	<b>13%</b>	<b>12%</b>	<b>7%</b>	<b>9%</b>



# Nota final

- ▶ La nota final se incrementará en **1 punto** a aquellos estudiantes que realicen:
  - ▶ **Todas** las pruebas de evaluación continua.
    - ▶ Todos los exámenes pequeños (**los 3**)
    - ▶ Todas las prácticas
  - ▶ Obtengan más de un 7 sobre 10 de calificación en la evaluación continua y al menos 4 puntos en el examen final.

Grupo ARCOS

**uc3m** | Universidad **Carlos III** de Madrid

# Estructura de Computadores

## Objetivos y presentación del curso

### Estructura de Computadores

Grado en Ingeniería Informática

Grado en Matemática aplicada y Computación

Doble Grado en Ingeniería Informática y Administración de Empresas



# ¿Por qué estudiar Estructura de Computadores? W. Stallings

El «IEEE/ACM Computer Curricula 2001» [JTF01], preparado por la *Joint Task Force* de currículo de computadores de la Sociedad de Computadores IEEE (*Institute of Electrical and Electronics Engineers*) y la ACM (*Association for Computing Machinery*), citan la arquitectura de computadores como uno de los temas troncales que debe estar en todos los currículos de todos los estudiantes de licenciatura e ingeniería informática. El informe dice lo siguiente:

«El computador está en el corazón de la informática. Sin él la mayoría de las asignaturas de informática serían hoy una rama de la matemática teórica. Para ser hoy un profesional en cualquier campo de la informática uno no debe ver al computador como una caja negra que ejecuta programas mágicamente. Todos los estudiantes de informática deben, en cierta medida, comprender y valorar los componentes funcionales de un computador, sus características, su funcionamiento y sus interacciones. También sus implicaciones prácticas. Los estudiantes necesitan comprender la arquitectura del computador para estructurar un programa de forma que este sea más eficiente en una máquina real. Seleccionando el sistema que se va a usar, debe ser capaz de comprender el compromiso entre varios componentes, como la velocidad del reloj de la CPU frente al tamaño de la memoria».

# ¿Por qué estudiar Estructura de Computadores? W. Stallings

En [CLEM00] se dan los siguientes ejemplos como razones para estudiar arquitectura de computadores:

1. Supóngase que un licenciado trabaja en la industria y se le pide seleccionar el computador con la mejor relación calidad precio para utilizarlo en una gran empresa. Comprender las implicaciones de gastar más en distintas alternativas, como una caché grande o una velocidad de reloj mayor, es esencial para tomar esta decisión.
2. Hay muchos procesadores que no forman parte de equipos PC o servidores, pero sí en sistemas embebidos. Un diseñador debe ser capaz de programar un procesador en C que esté embebido en algún sistema en tiempo real o sistema complejo, como un controlador electrónico de un coche inteligente. Depurar el sistema puede requerir utilizar un analizador lógico que muestre la relación entre las peticiones de interrupción de los sensores del sistema y el código máquina.
3. Los conceptos utilizados en arquitectura de computadores tienen aplicación en otros cursos. En particular, la forma en la que el computador ofrece un soporte arquitectural a los lenguajes de programación y funciones en principio propias del sistema operativo, refuerza los conceptos de estas áreas.