



# MESA REDONDA 1: Sistemas de Cómputo de Altas Prestaciones

## “Arquitecturas y Computación de Altas Prestaciones”

### ACAP



### *II Jornadas de Coordinación Docente y de Empresas*

*15 y 16 de Octubre de 2011*

**Patrocina:** Vicerrectorado para la Garantía de la Calidad



# Por qué necesitamos esta asignatura

- Cada día hay más **necesidades de recursos** para simulación **numérica**: gráficos, sistemas de predicción, de modelización etc.
- La **ejecución de grandes simulaciones numéricas** deben ir **acopladas a la arquitectura** para explotarla al máximo.
- Un Ingeniero de Computadores debe conocer tanto las **arquitecturas disponibles** para este tipo de computación como el proceso de **adaptación del cómputo a esas arquitecturas**.



# Por qué necesitamos esta asignatura

- La simulación numérica es útil por tres **razones**:
  - Disminuye los **costes** de construcción de sistemas reales
  - Aumenta la **productividad** disminuyendo el tiempo de desarrollo
  - Aumenta la **seguridad** de los sistemas reales
- Esta asignatura debe dar las nociones necesarias para saber **adaptar** los algoritmos de cómputo a una arquitectura paralela y los mecanismos básicos de **funcionamiento** de este tipo de arquitecturas.



## Localización

- Asignatura obligatoria de la especialidad de "Ingeniería de Computadores"
- Se impartirá en 3º del Grado de Informática (2º cuatrimestre)
- Duración: 6 créditos ECTS



## Situación

- Los alumnos abordan esta asignatura en su primer año de especialidad.
- Tienen nociones de:
  - Arquitectura de computadores
  - Estructura de computadores
  - Tecnología de computadores
  - Redes
  - Programación





## ¿Qué debe proporcionar?

- La capacidad de diseñar y construir sistemas digitales, incluyendo computadores, sistemas basados en microprocesador y **sistemas de comunicaciones**. (IC1)
- La capacidad de **analizar y evaluar** arquitecturas de computadores, incluyendo **plataformas paralelas y distribuidas**, así como desarrollar y optimizar software de para las mismas. (IC3)
- La **capacidad de diseñar e implementar software** de sistema y **de comunicaciones**. (IC4)



## ¿Cómo lo haremos?

- Justificando **la necesidad** de la Computación de Altas Prestaciones.
- Familiarizando al alumno con los **conceptos básicos** del paralelismo, de la computación paralela y de las arquitecturas paralelas.
- Enseñando al alumno a **paralelizar algoritmos** adaptándolos a la arquitectura que subyace.
- Proporcionando las herramientas para **medir las prestaciones** de un sistema paralelo.



# Temario teórico

- Tema 1: **Arquitecturas MIMD**
  - Clasificaciones de arquitecturas MIMD.
  - Evaluación de prestaciones: Isoeficiencia, Ley de Gustafson.
- Este tema situará al alumno en el **ámbito** de la asignatura.
- Establecerá **conceptos básicos** como tipos de paralelismo, prestaciones, ganancias, etc.
- Enseñará a **evaluar prestaciones** relativas al tiempo de ejecución paralelo.





# Temario teórico

- Tema 2: **Modelos de programación Paralela adaptados a la arquitectura**
  - Programación en Memoria Compartida
  - Programación en Memoria Distribuida
  - Modelos basados en datos
  - Modelos basados en hebras
  - Otros modelos
- Este tema establece los **principios de paralelización** de algoritmos que serán puestos en práctica en la parte práctica de la asignatura.



## Temario teórico

- Tema 3: **Redes de interconexión: Redes SAN**
  - Clasificación
  - Estructura general
  - Análisis de prestaciones
  - Diseño: niveles de servicio
  - Técnicas de conmutación
  - Control de flujo
  - Enrutamiento
- Este tema ayudará al alumno a conocer el **funcionamiento de los sistemas de comunicación** de las arquitecturas paralelas que deben dar servicio



# Temario Práctico

**Seminario 1:** Programación paralela I: **MPI.**

**Seminario 2:** Diseño de programas Paralelos: Manual y automático, **Ejemplos**

**Seminario 3:** Programación paralela II: **Java**

**Seminario 4:** Visita al **UGRGrid**

**Práctica 1:** Paralelización de un algoritmo base: MPI

**Práctica 2:** Análisis de Rendimiento de un algoritmo paralelo: Medida de prestaciones

**Práctica 3:** Programación basada en hebras: Java

**Práctica 4:** Programación distribuida en Java: Medida de prestaciones



# ¿Cuestion

