

## Guía docente

### 340384 - PACO-I5001 - Paralelismo y Concurrencia

Última modificación: 23/05/2025

**Unidad responsable:** Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú  
**Unidad que imparte:** 701 - DAC - Departamento de Arquitectura de Computadores.

**Titulación:** GRADO EN INGENIERÍA INFORMÁTICA (Plan 2018). (Asignatura obligatoria).

**Curso:** 2025      **Créditos ECTS:** 6.0      **Idiomas:** Catalán, Castellano

#### PROFESORADO

---

**Profesorado responsable:** Eva Marín Tordera

**Otros:** Eva Marín Tordera  
Farreras Esclusa, Montserrat

#### CAPACIDADES PREVIAS

---

Es recomendable haber cursado Arquitectura de Computadores y Sistemas Operativos.

#### COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

---

**Específicas:**

1. CEFB5. Conocimiento de la estructura, funcionamiento e interconexión de los sistemas informáticos, así como los fundamentos de su programación.
2. CEFC2. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.
3. CEFC8. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
4. CEFC9. Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.
5. CEFC14. Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

#### METODOLOGÍAS DOCENTES

---

Las clases de teoría se realizarán utilizando los medios disponibles en las aulas (pizarras, equipamiento multimedia) y están basadas en la exposición oral por parte del profesorado de los contenidos sobre la materia objeto de estudio (método expositivo). En algunos casos, se realizarán clases expositivas basadas en la participación e intervención de los estudiantes mediante actividades de corta duración en el aula, como son las preguntas directas, las exposiciones de los estudiantes sobre temas determinados o la resolución de problemas vinculados con el planteamiento teórico expuesto. También el profesor resolverá ejercicios en clase y propondrá ejercicios de la colección para que los estudiantes los preparen de forma autónoma. Estos ejercicios los resolverán en clase los estudiantes, individualmente o en grupo.

Las clases de grupo pequeño serán:

- Clases de laboratorio: se realizarán en las aulas informáticas del centro. El estudiante deberá llevar la práctica preparada (leer y entender el enunciado de la práctica a partir de un guión que se encontrará previamente en el campus digital), ya veces si así se indica deberá hacer un informe previo. Las prácticas serán individuales o en grupos de 2 personas.

## OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Los principales objetivos de esta asignatura son:

- Paradigmas de concurrencia, paralelismo y sistemas distribuidos: (cliente-servidor, distribución de la carga, tareas, etc.).
- Plataformas paralelas (arquitecturas de memoria compartida, memoria distribuida).
- Herramientas de ayuda al desarrollo de programas paralelos
- Programación y evaluación de programas paralelos (modelos de programación para las diferentes plataformas paralelas).
- Coherencia y consistencia de memoria. Comunicación, sincronización, condiciones de carrera, exclusión mutua, sección crítica, monitores, abrazo mortal.

## HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTE

Tipo	Horas	Porcentaje
Horas grupo grande	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00
Horas grupo pequeño	30,0	20.00

**Dedicación total:** 150 h

## CONTENIDOS

### 1. Introducción al paralelismo y la concurrencia

#### Descripción:

Necesidad de paralelismo. Paralelismo versus concurrencia. Problemas en el uso de la concurrencia: deadlock, LifeLock, Starvation, fairness, data races

#### Actividades vinculadas:

Actividad 1: Problemas tema 1

Actividad 2: Lab 1: Experimental setup, tools and programming model

#### Dedicación: 11h

Grupo grande/Teoría: 1h

Grupo mediano/Prácticas: 2h

Aprendizaje autónomo: 8h

### 2. Análisis de aplicaciones paralelas

#### Descripción:

Puede un cálculo dividirse en diferentes partes? Basándose en las tareas a realizar o basándose en los datos a repartir.

Habrà dependencia de datos entre las tareas? Cómo se resolverán?

Una buena descomposición determina el grado de paralelismo alcanzable

#### Actividades vinculadas:

Actividad 1: Problemas de análisis de aplicaciones paralelas

Actividad 2: Lab 1: Experimental setup, tools and programming model

Actividad 2: Lab 2: Brief tutorial on OpenMP programming model

#### Dedicación: 18h

Grupo grande/Teoría: 2h

Grupo mediano/Prácticas: 2h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 10h

### 3. Principios de programación paralela:descomposición en tareas

**Descripción:**

Identificación de patrones de concurrencia. Descomposición en tareas, granularidad y análisis de dependencias. Identificación de patrones de paralelismo: task paralelismos versus divide and conquer. Mecanismos para implementar la descomposición en tareas: thread creation and destruction, thread synchronization patterns, exclusión en el acceso a datos compartidos.

**Actividades vinculadas:**

Actividad 1: Problemas paralelismo tareas

Actividad 2: Lab 3: Analysis of parallel strategies: the computation of the Mandelbrot set

Actividad 2: Lab 4: Parallel Task Decomposition Implementation and Analysis

Actividad 4: Prueba de conocimiento

**Competencias relacionadas:**

I\_CEEFC14. CEEFC14. Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

I\_CEEFC9. CEEFC9. Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman.

I\_CEEFB5. CEEFB5. Conocimiento de la estructura, funcionamiento e interconexión de los sistemas informáticos, así como los fundamentos de su programación.

I\_CEEFC2. CEEFC2. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.

I\_CEEFC8. CEEFC8. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

**Dedicación:** 23h

Grupo grande/Teoría: 4h

Grupo pequeño/Laboratorio: 4h

Actividades dirigidas: 2h

Aprendizaje autónomo: 13h

### 4. Introducción a las arquitecturas (de memoria compartida) paralelas

**Descripción:**

Paralelismo dentro de un procesador (ILDP, DLP, TLP), multiprocesadores de memoria compartida, multiprocesadores de memoria distribuida

**Actividades vinculadas:**

Actividad 1: Problemas tema 4

**Dedicación:** 21h

Grupo grande/Teoría: 2h

Grupo mediano/Prácticas: 6h

Aprendizaje autónomo: 13h

## 5. Principios de programación paralela: descomposición en datos

### Descripción:

Descomposición de datos (geométrica versus estructura recursiva), organización del flujo de datos (regular versus irregular). Mecanismos para implementar la descomposición: process creation and destruction, process synchronization (barrier) and communications patterns (point-to-point communication, synchronous and Asynchronous communication)

### Actividades vinculadas:

Actividad 1: Problemas descomposición de datos

Actividad 2: Lab 5: Parallel Data Decomposition Implementation and Analysis

Actividad 3: Treball complementari

### Dedicación: 17h

Grupo grande/Teoría: 3h

Grupo mediano/Prácticas: 4h

Grupo pequeño/Laboratorio: 2h

Aprendizaje autónomo: 8h

## SISTEMA DE CALIFICACIÓN

Prueba conocimiento 1er parcial \* 0,2 + problemas \* 0,1 + Laboratorio \* 0,35 + trabajo complementario \* 0,1 + Prueba conocimiento 2º parcial \* 0,25 = 5

El segundo examen parcial puede ser 2º examen parcial con un peso del 25% o un examen final con un peso del 45% con el fin de recuperar el primer parcial (el alumno elige qué hacer). En este segundo caso la fórmula es:

Problemas \* 0,1 + Laboratorio \* 0,35 + trabajo complementario \* 0,1 + Prueba conocimiento Final \* 0,45 = 5

Es reevaluable la prueba de conocimiento final.

El trabajo complementario es opcional, si se hace cuenta un 10% de la nota, si no se hace, el examen 1er parcial tendrá un peso de un 25% y el segundo parcial un 30%, o si se hace el final tendría un peso de un 55%.

## NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Las actividades 1, 2 y 4 son presenciales.

La actividad 3 es no presencial, aunque puede haber una pequeña presentación en clase.

En las actividades que se realizan en grupo la calificación será la misma para todos los miembros del grupo

## BIBLIOGRAFÍA

### Básica:

- Mattson, Timothy G.; Sanders, Beverly A.; Massingill, Berna. Patterns for parallel programming [en línea]. Boston [etc.]: Addison-Wesley, 2005 [Consulta: 14/02/2024]. Disponible a: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?pg-origsite=primo&docID=7116386>. ISBN 0321228111.

## RECURSOS

### Material informático:

- Software a Boada. Conexión y software en boada.ac.upc.edu

### Otros recursos:

Se recomienda utilizar ordenador portátil con terminal