



Universidad  
Internacional  
de Valencia

# Guía didáctica

## **ASIGNATURA: *Sistemas Concurrentes y Distribuidos***

**Título:** *Grado en Ingeniería Informática*

**Materia:** *Programación de Computadores*

**Créditos:** 6 ECTS

**Código:** 19GIIN

# Índice

1. Organización general.....	3
1.1. Datos de la asignatura.....	3
1.2. Equipo docente .....	3
1.3. Introducción a la asignatura.....	3
1.4. Competencias y resultados de aprendizaje .....	4
2. Contenidos/temario .....	5
3. Metodología .....	6
4. Actividades formativas .....	7
5. Evaluación.....	8
5.1. Sistema de evaluación.....	8
5.2. Sistema de calificación .....	9
6. Bibliografía.....	9
6.1. Bibliografía de referencia .....	9
6.2. Bibliografía complementaria.....	10

# 1. Organización general

## 1.1. Datos de la asignatura

<b>MÓDULO</b>	<b>Común de la Rama Informática</b>
<b>MATERIA</b>	<b>Programación de Computadores</b>
<b>ASIGNATURA</b>	<i>Sistemas Concurrentes t Distribuidos</i> <b>6 ECTS</b>
<b>Carácter</b>	Obligatoria
<b>Curso</b>	Segundo
<b>Cuatrimestre</b>	Segundo
<b>Idioma en que se imparte</b>	Castellano
<b>Requisitos previos</b>	Se recomienda la superación de los contenidos y la adquisición de las materias de formación básica y en especial de las asignaturas Fundamentos de programación y Metodología de la programación
<b>Dedicación al estudio por ECTS</b>	<b>25 horas</b>

## 1.2. Equipo docente

<b>Profesor</b>	<b>Dr. Yudith C. Cardinale Villarreal</b> <b>Doctor en Computación</b> <a href="mailto:yudith.cardinale@campusviu.es">yudith.cardinale@campusviu.es</a>
-----------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------

## 1.3. Introducción a la asignatura

Los sistemas operativos modernos, así como las crecientes arquitecturas de múltiples núcleos, demandan herramientas de soporte a la programación de aplicaciones multi-tareas, que incluyen la comunicación por memoria compartida y el control de concurrencia (sincronización y exclusión mutua). Dependiendo de la técnica de programación multi- tareas usada (con procesos o con hilos), los programadores pueden seleccionar las herramientas apropiadas para el desarrollo de sus aplicaciones. Así mismo, cuando las aplicaciones multi-tareas se ejecutan en plataformas distribuidas, se requieren herramientas de comunicación y control de concurrencia (sincronización y exclusión mutua) para memoria distribuida. El objetivo de esta materia es dar a conocer los mecanismos de comunicación entre procesos, tanto en arquitecturas de memoria compartida como en arquitecturas de memoria distribuida, principalmente en ambientes Linux, con lenguaje C y Java.

V.04

## 1.4. Competencias y resultados de aprendizaje

### COMPETENCIAS GENERALES

- CG1 - Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
- CG2 - Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.
- CG4 - Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.
- CG5 - Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.
- CG8 - Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- CG10 - Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.

### COMPETENCIAS ESPECÍFICAS DE LA ASIGNATURA

- CE.1.- Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- CE.5.- Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.
- CE.6.- Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
- CE.7.- Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

CE.8.- Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CE.11.- Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.

CE.14.- Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

## RESULTADOS DE APRENDIZAJE

Al finalizar esta asignatura se espera que el estudiante sea capaz de:

RA.1.- Identificar las principales características de los distintos tipos de sistemas concurrentes que existen para comprender la importancia de la programación concurrente en las aplicaciones de hoy en día.

RA.2.- Describir los principales modelos de programación concurrente, paralela y distribuida.

RA.3.- Desarrollar algoritmos para sistemas basados en memoria compartida y para sistemas distribuidos que resuelvan problemas modelo en programación concurrente.

RA.4.- Usar las bibliotecas y plataformas estandarizadas para la implementación de programas concurrentes basados en memoria compartida y para sistemas distribuidos.

## 2. Contenidos/temario

### Unidad Competencial 1

- Conceptos básicos de programación concurrente: Elementos de hardware y software de un sistema de computación. Definición de concurrencia. Modelo de programación concurrente. Correctitud de programas concurrentes.
- Concepto de procesos: Manejo de procesos en ambientes Unix. Modelo de procesos. Filosofía de procesos en Unix. Jerarquía de procesos en Unix. Órdenes del shell para manipular procesos en Unix. Llamadas al sistema relacionadas con administración de procesos en Unix. Procesos huérfanos y zombies.
- Concepto de hilos: Manejo de hilos en ambientes Unix (en lenguaje C y en Java). Procesos multi-hilos. Librería Posix para manejar hilos en lenguaje C. Hilos en lenguaje Java.

## Unidad Competencial 2

- Problemas de exclusión mutua y sincronización en programación concurrente: Soluciones para exclusión mutua y sincronización. Ejemplos clásicos de exclusión mutua y sincronización. Problema de productores y consumidores. Problema de lectores y escritores. Problema de los filósofos.
- Mecanismos de exclusión mutua y sincronización: semáforos y monitores (en lenguaje C y en Java). Mecanismos de control de concurrencia de bajo nivel. Semáforos y mutex en Linux. Ejemplos de soluciones a los problemas clásicos de concurrencia usando semáforos y mutex. Mecanismos de control de concurrencia de alto nivel. Monitores en Java. Ejemplos de soluciones a los problemas clásicos de concurrencia usando monitores.

## Unidad Competencial 3

- Mecanismos de comunicación en memoria compartida: pipes y señales. Comunicación usando pipes en Unix. Pipes desde el shell. Pipes desde un programa en C. Comunicación usando señales en Unix.

## Unidad Competencial 4

- Mecanismos de comunicación en memoria distribuida: Sistemas distribuidos. Sockets en C y Java, RPC (Remote Procedure Call), RMI (Remote Method Invocation).

# 3. Metodología

La metodología de la Universidad Internacional de Valencia (VIU) se caracteriza por una apuesta decidida en un modelo de carácter e-presencial. Así, siguiendo lo estipulado en el calendario de actividades docentes del Título, se impartirán en directo un conjunto de sesiones, que, además, quedarán grabadas para su posterior visionado por parte de aquellos estudiantes que lo necesitasen. En todo caso, se recomienda acudir, en la medida de lo posible, a dichas sesiones, facilitando así el intercambio de experiencias y dudas con el docente.

En lo que se refiere a las metodologías específicas de enseñanza-aprendizaje, serán aplicadas por el docente en función de los contenidos de la asignatura y de las necesidades pedagógicas de los estudiantes. De manera general, se impartirán contenidos teóricos y, en el ámbito de las clases prácticas se podrá realizar la resolución de problemas, el estudio de casos y/o la simulación.

Por otro lado, la Universidad y sus docentes ofrecen un acompañamiento continuo al estudiante, poniendo a su disposición foros de dudas y tutorías para resolver las consultas de carácter académico que el estudiante pueda tener. Es importante señalar que resulta

fundamental el trabajo autónomo del estudiante para lograr una adecuada consecución de los objetivos formativos previstos para la asignatura.

## 4. Actividades formativas

Durante el desarrollo de cada una de las asignaturas se programan una serie de actividades de aprendizaje que ayudan a los estudiantes a consolidar los conocimientos trabajados.

A continuación, se relacionan las actividades que forman parte de la asignatura:

### 1. Actividades de carácter teórico

Se trata de un conjunto de actividades guiadas por el profesor de la asignatura destinadas a la adquisición por parte de los estudiantes de los contenidos teóricos de la misma. Estas actividades, diseñadas de manera integral, se complementan entre sí y están directamente relacionadas con los materiales teóricos que se ponen a disposición del estudiante (manual, SCORM y material complementario). Estas actividades se desglosan en las siguientes categorías:

- a. Clases expositivas
- b. Sesiones con expertos en el aula
- c. Observación y evaluación de recursos didácticos audiovisuales
- d. Estudio y seguimiento de material interactivo

### 2. Actividades de carácter práctico

Se trata de un conjunto de actividades guiadas y supervisadas por el profesor de la asignatura vinculadas con la adquisición por parte de los estudiantes de los resultados de aprendizaje y competencias de carácter más práctico. Estas actividades, diseñadas con visión de conjunto, están relacionadas entre sí para ofrecer al estudiante una formación completa e integral.

### 3. Tutorías

Se trata de sesiones, tanto de carácter síncrono como asíncrono (e-mail), individuales o colectivas, en las que el profesor comparte información sobre el progreso académico del estudiante y en las que se resuelven dudas y se dan orientaciones específicas ante dificultades concretas en el desarrollo de la asignatura.

### 4. Trabajo autónomo

Se trata de un conjunto de actividades que el estudiante desarrolla autónomamente y que están enfocadas a lograr un aprendizaje significativo y a superar la evaluación de la asignatura. La realización de estas actividades es indispensable para adquirir las competencias y se encuentran entroncadas en el aprendizaje autónomo que consagra la actual ordenación de enseñanzas universitarias. Esta actividad, por su definición, tiene carácter asíncrono.

## 5. Prueba objetiva final

Como parte de la evaluación de cada una de las asignaturas (a excepción de las prácticas y el Trabajo fin de título), se realiza una prueba (examen final). Esta prueba se realiza en tiempo real (con los medios de control antifraude especificados) y tiene como objetivo evidenciar el nivel de adquisición de conocimientos y desarrollo de competencias por parte de los estudiantes. Esta actividad, por su definición, tiene carácter síncrono.

# 5. Evaluación

## 5.1. Sistema de evaluación

El Modelo de Evaluación de estudiantes en la Universidad se sustenta en los principios del Espacio Europeo de Educación Superior (EEES), y está adaptado a la estructura de formación virtual propia de esta Universidad. De este modo, se dirige a la evaluación de competencias.

Sistema de Evaluación	Ponderación
<b>Portafolio*</b>	<b>40 %</b>
Entrega de informes de problemas y ejercicios	10%
Planteamiento, estudio, análisis y resolución de casos	10 %
Informes o memorias de prácticas de laboratorio	15 %
Participación activa en los debates, foros y otros medios	5 %
Sistema de Evaluación	Ponderación
<b>Prueba final*</b>	<b>60 %</b>
La prueba final consistirá en un examen tipo test con preguntas de verdadero y falso, selección simple y desarrollo de respuestas cortas (análisis de un trozo de código, desarrollo de una idea, etc.), que mostrará el desarrollo de las competencias específicas y los resultados del aprendizaje trabajados en cada Unidad Competencial.	

**\*Es requisito indispensable para superar la asignatura aprobar cada apartado (portafolio y prueba final) con un mínimo de 5 para ponderar las calificaciones.**

Los enunciados y especificaciones propias de las distintas actividades serán aportados por el docente, a través del Campus Virtual, a lo largo de la impartición de la asignatura.

Atendiendo a la Normativa de Evaluación de la Universidad, se tendrá en cuenta que la utilización de **contenido de autoría ajena** al propio estudiante debe ser citada adecuadamente en los trabajos entregados. Los casos de plagio serán sancionados con suspenso (0) de la

actividad en la que se detecte. Asimismo, el uso de **medios fraudulentos durante las pruebas de evaluación** implicará un suspenso (0) y podrá implicar la apertura de un expediente disciplinario.

## 5.2. Sistema de calificación

La calificación de la asignatura se establecerá en los siguientes cálculos y términos:

Nivel de aprendizaje	Calificación numérica	Calificación cualitativa
Muy competente	9,0 - 10	Sobresaliente
Competente	7,0 - 8,9	Notable
Aceptable	5,0 -6,9	Aprobado
Aún no competente	0,0 -4,9	Suspenso

Sin detrimento de lo anterior, el estudiante dispondrá de una **rúbrica simplificada** en el aula que mostrará los aspectos que valorará el docente, como así también los **niveles de desempeño que tendrá en cuenta para calificar las actividades vinculadas a cada resultado de aprendizaje**.

La mención de «**Matrícula de Honor**» podrá ser otorgada a estudiantes que hayan obtenido una calificación igual o superior a 9.0. Su número no podrá exceder del cinco por ciento de los estudiantes matriculados en una materia en el correspondiente curso académico, salvo que el número de estudiantes matriculados sea inferior a 20, en cuyo caso se podrá conceder una sola «Matrícula de Honor».

## 6. Bibliografía

### 6.1. Bibliografía de referencia

- Glass, G., & Ables, K. (2003). *“UNIX for programmers and users”*. 3rd Ed. New York: Prentice Hall.
- Herbert, B., & Tanenbaum, A. S. (2014). *“Modern operating systems”*. 4Th Ed. Pearson.
- Kernighan, B. W., & Ritchie, D. M. (1991). *“El lenguaje de programación C”*. Pearson Educación.
- Silberschatz, A., Galvin, P. B., & Gagne, G. (2014). *“Operating system concepts essentials”*. John Wiley & Sons, Inc.
- Stallings, W., & Manna, M. M. (2015). *“Operating systems: internals and design principles”*. Pearson.

## 6.2. Bibliografía complementaria

- Steven, R., & Robbins, K. A. (2000). *“UNIX Programación práctica. Guía para la Concurrencia, la Comunicación y los Multihilos”*. Prentice Hall.
- Vahalia, U. (2008). *“UNIX internals: the new frontiers”*. Pearson Education India.
- Lewis, B., & Berg, D. J. (2000). *“Multithreaded programming with Javatechnology”*. Prentice Hall Professional.
- Lewis, B., & Berg, D. J. (1998). *“Multithreaded programming with Pthreads”*. Prentice-Hall, Inc..