

## **FICHA DE ASIGNATURA**

**Título:** Sistemas Concurrentes y Distribuidos

**Descripción:** 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.

**Carácter:** Obligatoria

**Créditos ECTS:** 6

**Contextualización:** El programa de esta asignatura brinda una serie de contenidos teóricos y prácticos que permitirá a los estudiantes identificar las principales características de los sistemas concurrentes, describir los principales modelos de programación concurrente, paralela y distribuida, así como desarrollar algoritmos para sistemas basados en memoria compartida y para sistemas distribuidos, que resuelvan problemas modelo en programación concurrente.

**Modalidad:** Online

**Temario:**

- 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.
- 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.

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

### **Competencias:**

**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.

**CG3.** 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.

**CG4.** 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.

**CG5.** 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.

**CG6.** Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

**CG7.** Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

**CG8.** Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

**CG9.** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

**CG10.** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

**CE1.** Comprensión y dominio de los conceptos básicos de la programación concurrente, paralela y distribuida que le permitan resolver problemas informáticos en concordancia con las herramientas de software y hardware que disponga el estudiante.

**Actividades Formativas:**

<b>Actividad Formativa</b>	<b>Horas</b>	<b>Presencialidad</b>
Clases expositivas	60	60
Resolución de ejercicios prácticos	80	30
Prácticas de laboratorios virtuales	100	20
Tutorías	60	0
Trabajo autónomo	300	0

**Metodologías docentes:**

- Clases teóricas impartidas como lecciones magistrales o exposiciones, en las que además de presentar el contenido de la asignatura, se explican los conceptos fundamentales y se desarrolla el contenido teórico.
- Colección de tareas que el alumnado llevará a cabo a lo largo de toda la asignatura, entre las que podemos encontrar: análisis de casos, resolución de problemas, prácticas de laboratorios, comentarios críticos de textos, análisis de lecturas, etc.
- Sesiones periódicas entre el profesorado y el alumnado para la resolución de dudas, orientación, supervisión, etc.
- Trabajo tanto individual como grupal para la lectura crítica de la bibliografía, estudio sistemático de los temas, reflexión sobre problemas planteados, resolución de actividades propuestas, búsqueda, análisis y elaboración de información, investigación e indagación, así como trabajo colaborativo basado en principios constructivistas.

**Sistema de Evaluación:**

<b>Sistemas de evaluación</b>	<b>Ponderación mínima</b>	<b>Ponderación máxima</b>
Entrega de informes de problemas y ejercicios	5.0	20.0
Planteamiento, estudio, análisis y resolución de casos	0.0	10.0
Informes o memorias de prácticas de laboratorio	10.0	15.0
Trabajos o proyectos desarrollados en grupo o de forma individual	0.0	40.0
Participación activa en los debates, foros y otros medios	0.0	5.0
Evaluación final	40.0	60.0

**Normativa específica:** No aplica

**Bibliografía:**

- 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.
- 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*. India: Pearson Education.
- Lewis, B., & Berg, D. J. (2000). *Multithreaded programming with Java technology*. Prentice Hall Professional.
- Lewis, B., & Berg, D. J. (1998). *Multithreaded programming with Pthreads*". Prentice-Hall, Inc..