



Guía Didáctica - GRADO

ASIGNATURA: **Sistemas Concurrentes y Distribuidos**

Título: **Grado en Ingeniería Informática**

Módulo: **Fundamentos de Informática**

Créditos: **6 ECTS**

Código: **19GIIN**

Índice

1. Organización general	3
1.1. Datos de la asignatura	3
1.2. Introducción a la asignatura	3
1.3. Competencias y resultados de aprendizaje	4
2. Contenidos/temario	6
3. Actividades Formativas	7
4. Metodologías Docentes	8
5. Evaluación	9
5.1. Sistema de evaluación.....	9
5.2. Sistema de Calificación.....	10
6. Bibliografía	11

1. Organización general

1.1. Datos de la asignatura

MÓDULO	Fundamentos de Informática
MATERIA	Programación de computadores
ASIGNATURA	Sistemas Concurrentes y Distribuidos 6 ECTS
Carácter	Obligatorio
Curso	Segundo
Cuatrimestre	Segundo
Idioma en que se imparte	Castellano
Requisitos previos	<ul style="list-style-type: none"> • Superación de los contenidos y la adquisición de las materias de formación básica, en especial de las asignaturas Fundamentos de programación y Metodología de la programación. • Superación de los contenidos de la asignatura Sistemas operativos.
Dedicación al estudio recomendada por ECTS	25 horas

1.2. 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.

1.3. Competencias y resultados de aprendizaje

COMPETENCIAS BÁSICAS

CB1. 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.

CB2. 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.

CB3. 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.

CB4. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

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.

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.

COMPETENCIAS ESPECÍFICAS DE LA ASIGNATURA

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.

RESULTADOS DE APRENDIZAJE

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

RA1. 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.

RA2. Describir los principales modelos de programación concurrente, paralela y distribuida.

RA3. Desarrollar algoritmos para sistemas basados en memoria compartida y para sistemas distribuidos que resuelvan problemas modelo en programación concurrente.

RA4. 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. Actividades Formativas

ACTIVIDAD FORMATIVA	HORAS	PRESENCIALIDAD
Clases expositivas	60	60
Resolución de ejercicios prácticos	80	30
Prácticas de laboratorios virtuales	100	20
Tutorías	60	0
Trabajo Autónomo	300	0

4. 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.

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.

Es requisito indispensable aprobar el portafolio y la prueba final con un mínimo de 5 para ponderar las calificaciones.

Sistema de Evaluación	Ponderación
Portafolio*	50 %
Colección de tareas realizadas por el alumnado y establecidas por el profesorado. La mayoría de las tareas aquí recopiladas son el resultado del trabajo realizado dirigido por el profesorado en las actividades, tutorías, etc. Esto permite evaluar, además de las competencias conceptuales, otras de carácter más práctico, procedimental o actitudinal.	
Sistema de Evaluación	Ponderación
Prueba final*	50 %
La realización de una prueba cuyas características son definidas en cada caso por el correspondiente profesorado.	

Unidad competencial	Portafolio	Prueba Final
UC 1 - 30%	15%	15%
UC 2 - 30%	15%	15%
UC 3 - 25%	10%	10%
UC 4 - 15%	10%	10%
TOTALES	50%	50%

***Es requisito indispensable para superar la asignatura aprobar cada apartado (portafolio y prueba final).**

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 Competencia	Calificación Oficial	Etiqueta Oficial
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

El nivel de competencia en cada una de las actividades realizadas se medirá, teniendo en cuenta **criterios generales derivados de la consecución de los resultados de aprendizaje**, que en términos generales y en función de la adecuación en el planteamiento de los contenidos generales y contenidos específicos, valorarán por norma general y en trabajos escritos, la corrección de la estructura formal y organización del discurso (semántica, sintaxis y léxico) valorándose además la originalidad, creatividad y argumentación de las intervenciones utilizando referencias bibliográficas.

Sin detrimento de lo anterior, el alumnado dispondrá de una **rúbrica simplificada** 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**.

6. Bibliografía

Bibliografía Básica:

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.

Bibliografía Opcional

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 Java technology”. Prentice Hall Professional.

Lewis, B., & Berg, D. J. (1998). “Multithreaded programming with Pthreads”. Prentice-Hall, Inc.