

PLAN DOCENTE DE LA ASIGNATURA

Curso académico: 2026/2027

Identificación y características de la asignatura						
Código	501284					
Denominación (español)	Programación Concurrente y Distribuida					
Denominación (inglés)	Concurrent and Distributed Programming					
Titulaciones	Grado en Ing. Informática en Ingeniería de Computadores Grado en Ing. Informática en Ingeniería del Software					
Centro	Escuela Politécnica					
Módulo	Común a la rama de Informática					
Materia	Programación					
Carácter	Obligatorio	ECTS	6	Semestre	4	
Profesorado						
Nombre		Despacho		Correo-e		
Fernando Sánchez Figueroa		I.4 (Despacho 56)		fernando@unex.es		
Álvaro Rubio Largo		I.2 (Despacho 15)		arl@unex.es		
Daniel Flores Martín		I.2 (Despacho 15)		dfloresm@unex.es		
Área de conocimiento	Lenguajes y Sistemas Informáticos					
Departamento	Ingeniería de Sistemas Informáticos y Telemáticos					
Profesor/a coordinador	Fernando Sánchez Figueroa					
Competencias						
COMPETENCIAS BÁSICAS (Competencias básicas establecidas para Grado en el Anexo I 3.2 del RD 861/2010)						
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

CG02: Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la Informática de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del Anexo II de la resolución antes mencionada para las tecnologías específicas de Ingeniería del Software e Ingeniería de Computadores.

CG04: 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 el apartado 5 del Anexo II de la resolución antes mencionada para las tecnologías específicas de Ingeniería del Software e Ingeniería de Computadores.

CG08: 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.

CG09: Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

COMPETENCIAS ESPECÍFICAS

(Según los planes de estudio aprobados, esta asignatura debe cubrir, total o parcialmente, las siguientes competencias específicas y sus resultados de aprendizaje)

CI11: 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.

CI14: Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

COMPETENCIAS TRANSVERSALES

(Según los planes de estudio aprobados y los acuerdos de la comisión de calidad de las titulaciones, esta asignatura debe cubrir, total o parcialmente, las siguientes competencias transversales y sus resultados de aprendizaje en un nivel intermedio)

CT09: Capacidad de trabajo en grupo.

CT16: Capacidad para adaptarse a nuevas situaciones y cambios.

La competencia transversal CT09 y sus resultados de aprendizaje se complementan en la asignatura "Inteligencia Artificial y Sistemas Inteligentes", también del 4º semestre.

Contenidos

Concurrencia y distribución. Primitivas de sincronización. Desarrollo de aplicaciones concurrentes y distribuidas.

Temario

TEMA 1. CONCEPTOS FUNDAMENTALES DE LA PROGRAMACIÓN CONCURRENTE

1.1 Introducción

1.2 Concepto de programación concurrente

- 1.3 Procesos y *threads*
- 1.4 Beneficios de la programación concurrente
- 1.5 Concurrencia y arquitecturas hardware
- 1.6 Características de los sistemas concurrentes
- 1.7 Problemas inherentes a la programación concurrente
- 1.8 Corrección de programas concurrentes
- 1.9 Creación avanzada de *threads*

Descripción de las actividades prácticas: Creación de *threads*. Medición del rendimiento de programas monohilo-multihilo. Definición del proyecto de la asignatura.

TEMA 2. PROBLEMAS DE SINCRONIZACIÓN. PRIMERAS SOLUCIONES

- 2.1 Tipos de sincronización
- 2.2 Solución a la condición de sincronización
- 2.3 Soluciones a la exclusión mutua
 - 2.3.1 Soluciones Software
 - 2.3.2 Soluciones Hardware
 - 2.3.3 Otras soluciones: deshabilitación de interrupciones

Descripción de las actividades prácticas: Identificación de los problemas de la programación concurrente mediante la realización de ejercicios.

TEMA 3. MECANISMOS DE SINCRONIZACIÓN DE HILOS

- 3.1 Variables atómicas
- 3.2 Cerrojos
- 3.3 Primitivas propias de Java
- 3.4 Monitores
- 3.5 Semáforos
- 3.6 Estructuras de datos concurrentes
- 3.7 Mecanismos avanzados de sincronización

Descripción de las actividades prácticas: Ejercicios con los diferentes mecanismos. Inclusión en el proyecto de la asignatura.

TEMA 4. PROGRAMACIÓN REACTIVA

- 4.1 Streams paralelos
- 4.2 Observables y Observadores
- 4.3 Concurrencia en programación reactiva

Descripción de las actividades prácticas: Ejercicios con programación reactiva. Inclusión en el proyecto de la asignatura.

TEMA 5. MECANISMOS DE PASO DE MENSAJE

- 5.1 Introducción
- 5.2 Características de los mecanismos de paso de mensaje
- 5.3 Paso de mensaje síncrono
- 5.4 Espera Selectiva
- 5.5 Paso de mensaje asíncrono

5.6 Invocación remota

Descripción de las actividades prácticas: Ejercicios con los diferentes tipos de paso de mensaje.

TEMA 6. PLATAFORMAS DE OBJETOS DISTRIBUIDOS

6.1 Plataformas de objetos distribuidos. Panorámica

6.2 Selección de plataforma

6.3 Plataforma gRPC

Descripción de las actividades prácticas: Ejercicios con plataformas de objetos distribuidos. Inclusión en el proyecto de la asignatura.

TEMA 7. PARADIGMAS Y PROBLEMAS DE LA PROGRAMACIÓN DISTRIBUIDA

7.1 Introducción

7.2 Paradigmas y arquitecturas de programación distribuida

7.3 Técnicas básicas de programación distribuida

7.4 Problemas básicos en la programación distribuida

Actividades formativas

Horas de trabajo del alumno/a por tema		Horas Gran grupo	Actividades prácticas				Actividad de seguimiento	No presencial
Tema	Total	GG	CH	L	O	S	TP	EP
1	8,5	3		3			0	2,5
2	10,5	3		1,5			0	6
3	36	10		6			0	24
4	20	6		3			1	16
5	9	3		1,5			0	6
6	30,5	5,5		6			0,5	24
7	13	4		0			0	10
Evaluación	4,5	3		1,5			0	0
TOTAL	150	37,5		22,5			1,5	88,5

GG: Grupo Grande (85 estudiantes).

CH: Actividades de prácticas clínicas hospitalarias (7 estudiantes)

L: Actividades de laboratorio o prácticas de campo (15 estudiantes)

O: Actividades en sala de ordenadores o laboratorio de idiomas (20 estudiantes)

S: Actividades de seminario o de problemas en clase (40 estudiantes).

TP: Tutorías Programadas (seguimiento docente, tipo tutorías ECTS).

EP: Estudio personal, trabajos individuales o en grupo, y lectura de bibliografía.

Metodologías docentes

La asignatura "Programación concurrente y distribuida" busca la participación activa y continuada de los estudiantes, quienes deberán hacer frente a nuevos retos que se irán proponiendo a lo largo de la asignatura, y donde se hará un uso intensivo del Campus Virtual.

A continuación, se detallan algunas de las actividades formativas que se plantearán a lo largo del curso para alcanzar los objetivos de aprendizaje de la asignatura. Aunque cada actividad sólo se detalla dentro de una modalidad (presenciales en grupo grande, presenciales en laboratorio, tutorías ECTS y no presenciales), algunas de ellas se desarrollarán en varias pudiendo, éstas, ser realizadas de forma individual y/o en grupo.

PRESENCIALES EN GRUPO GRANDE

Orientadas principalmente a la adquisición de los conceptos teóricos de la asignatura, en estas actividades se combinan las clases expositivas con la resolución de problemas individualmente y/o en grupo con metodologías activas de aprendizaje.

PRESENCIALES EN LABORATORIO

Las sesiones de laboratorio estarán a disposición de los alumnos antes del inicio de cada sesión. Cada sesión dispone de un guion que contiene los objetivos y los trabajos que se deben desarrollar. El estudiante deberá realizar parte de alguna de las sesiones previamente y de manera autónoma, de manera que en el momento de la sesión presencial de laboratorio sea capaz de implantar la solución software real. En ese caso, se seguirá una metodología de *Flipped Classroom*. Los grupos de laboratorio tendrán un número máximo de 20 alumnos.

TUTORÍAS ECTS

Las actividades formativas que se plantean en este bloque están orientadas, principalmente, a realizar el seguimiento de la adquisición de las competencias transversales. Mediante una estrategia de roles, se plantearán diversos problemas que el grupo debe ir resolviendo a lo largo del curso. Los requisitos de cada uno de estos problemas son cambiantes, de manera que el grupo debe hacer frente a las nuevas situaciones y cambios que requieren los nuevos requisitos. Se hará uso de rúbricas para determinar el grado de consecución de las actividades propuestas. Cada grupo de laboratorio se dividirá en 2 o 3 grupos ECTS cada uno.

NO PRESENCIALES

Dentro de las actividades no presenciales planteadas se incluyen diversas estrategias que complementan el aprendizaje autónomo del estudiante. Entre ellas se encuentran los videos y actividades preparatorias de las clases, siguiendo el enfoque de aula invertida (*flipped classroom*), el estudio individual, y las reuniones de grupo. También se considera la búsqueda de información, la elaboración de preguntas tipo test, el seguimiento de problemas resueltos, el acceso a la documentación disponible en el aula virtual, así como la comunicación con profesores y compañeros a través de foros. Finalmente, se contemplan los cuestionarios de evaluación y autoevaluación como herramientas para reforzar y comprobar los conocimientos adquiridos. Cuestionarios de evaluación y autoevaluación

Resultados de aprendizaje

ASOCIADOS A LAS COMPETENCIAS BÁSICAS Y ESPECÍFICAS

- Puede utilizar de manera eficaz un entorno de programación que incluya herramientas de edición, compilación, depuración y documentación de programas.
- Justifica la utilización de distintos paradigmas de programación y plataformas de desarrollo de software en un determinado contexto.

- Busca, analiza, sintetiza y critica nueva información para aprender nuevos lenguajes, algoritmos, técnicas, paradigmas y metodologías de programación aplicables a distintas áreas, teniendo como objetivo la actualización continua de los conocimientos y competencias.
- Conoce las principales primitivas relacionadas con la concurrencia y las aplica en el diseño de este tipo de sistemas.
- Conoce los conceptos fundamentales sobre sistemas de computación distribuida y sus distintas aplicaciones.

ASOCIADOS A LAS COMPETENCIAS TRANSVERSALES

- **CT09:**
 - Conoce las normas básicas de trabajo en equipo, colaboración, compromiso y responsabilidad y las técnicas básicas de trabajo
 - Conoce y aplica técnicas básicas de trabajo en equipos que trabajan de forma presencial o virtual.
 - Trabaja de manera eficiente como parte integrante o liderando equipos unidisciplinarios o multidisciplinares.
 - Contribuye al trabajo del grupo y favorece la buena comunicación, pudiendo desempeñar distintas funciones dentro del grupo.
 - Participa en el establecimiento de planes de trabajo equilibrados y efectivos, y evalúa su ejecución.
- **CT16:**
 - Identifica las situaciones de cambio.
 - Elabora las estrategias para abordar la problemática implicada por la nueva situación, aceptando ser flexible y estando dispuesto a cambiar las propias ideas ante una nueva información o vivencia contraria.

Sistemas de evaluación

Se aplicará el sistema de calificaciones vigente en el RD 1125/2003, artículo 5º y la RESOLUCIÓN de 26 de octubre de 2020 de la Universidad de Extremadura, por la que se ejecuta el Acuerdo adoptado por el Consejo de Gobierno por el que se aprueba la normativa de evaluación de las Titulaciones oficiales de Grado y Máster de la Universidad de Extremadura (DOE del 3-11-2020).

Para la evaluación de la asignatura existirán dos modalidades diferentes: evaluación continua y evaluación final. La elección entre el sistema de evaluación continua o el sistema de evaluación por prueba final de carácter global corresponde al estudiante.

Durante las primeras cuatro semanas del curso, el estudiante podrá elegir, a través del Campus Virtual, la modalidad con la que quiere ser evaluado para cada convocatoria (ordinaria/extraordinaria). En caso de ausencia de solicitud expresa por parte del estudiante, la modalidad asignada será la de evaluación continua.

CRITERIOS DE EVALUACIÓN PARA LA MODALIDAD EVALUACIÓN CONTINUA

Para aprobar la asignatura el estudiante deberá:

- Superar la evaluación de los conocimientos teóricos y prácticos con los requisitos mínimos propuestos en cada uno de los bloques que se detallan a continuación.
- La puntuación de cada bloque se calculará sobre 10.

Bloque 1: Teórico

- Este bloque es recuperable y su calificación será del 80% de la calificación total en la asignatura.
- Es obligatorio superar este bloque con una nota mínima de 4 para aprobar la asignatura.
- La evaluación de este bloque será mediante:
 - la realización de un test final.

Bloque 2: Práctico

- El alumno desarrollará un proyecto durante el curso.
- Este bloque es recuperable y su calificación será del 20% de la calificación total en la asignatura.
- Es obligatorio superar este bloque con una nota mínima de 4 para aprobar la asignatura.
- La evaluación de este bloque será mediante:
 - una prueba práctica final relacionada con el proyecto que podrá ser escrita o en ordenador (o mezcla de ambas) y que tiene una ponderación del 80% en este bloque.
 - seguimiento en clase del trabajo del alumno con revisiones parciales del proyecto y que tiene una ponderación del 20% en este bloque.

La copia o el plagio demostrados en cualquier actividad supone una nota final de SUSPENSO (0) en la convocatoria para todos los implicados, además de las actuaciones legales indicadas según la normativa vigente.

CRITERIOS EVALUACIÓN PARA LA MODALIDAD EVALUACIÓN FINAL

Para aprobar la asignatura el estudiante deberá:

- Superar la evaluación de los conocimientos teóricos y prácticos con los requisitos mínimos propuestos en cada uno de los bloques que se detallan a continuación.
- La puntuación de cada bloque se calculará sobre 10.

Bloque 1: Teórico

- Este bloque es recuperable y su calificación será del 80% de la calificación total en la asignatura.
- Es obligatorio superar este bloque con una nota mínima de 4 para aprobar la asignatura.
- La evaluación de este bloque será mediante un test final.

Bloque 2: Práctico

- Este bloque es recuperable y su calificación será del 20% de la calificación total en la asignatura.
- Es obligatorio superar este bloque con una nota mínima de 4 para aprobar la asignatura.
- La evaluación de este bloque será mediante:

- una prueba práctica final que podrá ser escrita o en ordenador (o mezcla de ambas).

La copia o el plagio demostrados en cualquier actividad supone una nota final de SUSPENSO (0) en la convocatoria para todos los implicados, además de las actuaciones legales indicadas según la normativa vigente.

Bibliografía (básica y complementaria)

BIBLIOGRAFÍA BÁSICA

- **[Ben90]** M. Ben-Ari. Principles of concurrent and Distributed Programming. Prentice-Hall
- **[Bur93]** A. Burns, G.L. Davies. Concurrent Programming. Addison-Wesley
- **[Fer12]** Javier Fernández González. Java Concurrency Cookbook. Editorial Packt Publishing Ltd.
- **[Fer16]** Javier Fernández González. Mastering Concurrency Programming with Java 8. Editorial Packt Publishing Ltd
- **[Gal15]** Ricardo Galli. Principios y Algoritmos de Concurrencia
- **[Har98]** S. J. Hartley. Concurrent Programming. The Java Programming Language. Oxford University Press.
- **[Ray92]** M. Raynal. Distributed algorithms and protocols. Ed. John Wiley & Sons.
- **[Mag99]** J. Magee, J. Kramer. Concurrency. State model & Java Programs. John Wiley & Sons
- **[Oak97]** S. Oaks, H. Wong. Java threads. O'Reilly & Associates
- **[Pal03]** J.T. Palma, M.C. Garrido, F. Sánchez, A. Quesada. Programación Concurrente. Editorial Thomson-Paraninfo (www.paraninfo.es)
- **[Per90]** J.E. Pérez. Programación Concurrente. Editorial Rueda.
- **[Sam20]** Nick Samoylov, Thomas Nield. Learning RxJava Second Edition. Editorial Packt Publishing Ltd

BIBLIOGRAFÍA COMPLEMENTARIA

- **[And91]** G.R. Andrews. Concurrent Programming. Principle and Practice. Addison-Wesley
- **[Axf89]** T. Axford. Concurrent Programming. Fundamental Techniques for Real-Time and Parallel Software Design. Editorial Wiley. Series en Parallel Computing
- **[Bac98]** J. Bacon. Concurrent Systems. Operating Systems, Database and Distributed Systems: An Integrated Approach. Addison Wesley
- **[Bev97]** J. Beveridge, R. Wiener. Multithreading Applications in Win32. Addison-Wesley
- **[Bus88]** D. Bustard, J. Elder, J. Welsh. Concurrent Program Structures. Prentice Hall International Series in Computer Science
- **[But97]** D. R. Butenhof. Programming with POSIX Threads. Addison-Wesley Professional Computing Series
- **[Dea00]** D. Lea. Programación Concurrente en Java. Principios y patrones de diseño. Addison Wesley
- **[Dei90]** H.M. Deitel. Sistemas Operativos. Addison-Wesley Iberoamericana, 2ª edición

- **[Geh88]** N. Gehani, A.D. McGettrick. Concurrent Programming. International Computer Science Series. Addison-Wesley
- **[Hol00]** Allen Holub. Taming Java Threads. Apress
- **[Hyd99]** Paul Hyde. Java Thread Programming. SAMS (división de MacMillan Computer Publishing)
- **[Lew00]** B. Lewis, D. J. Berg. Multithreaded programming with Java Technology. Sun Microsystems Press
- **[Lyn96]** Nancy Lynch, Distributed Algorithms, Ed. Morgan Kaufmann Publishers, 1996
- **[Mil94]** M. Milenkovic. Sistemas Operativos. Concepto y diseño. McGraw-Hill, 2ª edición
- **[Nic96]** B. Nichols, D. Buttler, J. Proulx. Pthreads Programming. O'Reilly
- **[Pet93]** J.L. Peterson, Abraham Silberschatz. Sistemas Operativos. Editorial Reverté

Otros recursos y materiales docentes complementarios

RECURSOS VIRTUALES

- Aula virtual de la asignatura

MEDIOS MATERIALES UTILIZADOS

- Pizarra
- Cañón de video
- Ordenador
- Internet

MATERIALES Y RECURSOS UTILIZADOS

Los materiales y recursos utilizados estarán disponibles en el espacio reservado para la asignatura en el Campus Virtual. Concretamente los alumnos dispondrán de:

- Trasparencias para cada tema del programa
- Guiones de las sesiones de laboratorio
- Foros de preguntas y respuestas
- Tablón de anuncios de novedades
- Glosarios de términos, palabras claves
- Baterías de preguntas de test
- Conjunto de referencias web relacionadas con la programación concurrente y distribuida
- Tests de autoevaluación de contenidos
- Tareas virtuales para la entrega de problemas propuestos
- Videotutoriales con conceptos teóricos y resolución de ejercicios