



Universidad
de Alcalá

GUÍA DOCENTE

Programación Avanzada

Grado en
Ingeniería Informática (GII)
Ingeniería de Computadores (GIC)
Ingeniería Informática y Administración y Dirección de
Empresas (GII-ADE)

Universidad de Alcalá

Curso Académico 2024/2025

2º Curso - 2º Cuatrimestre (GII+GIC)

3^{er} Curso - 2º Cuatrimestre (GII-ADE)

GUÍA DOCENTE

Nombre de la asignatura:	Programación Avanzada
Código:	780014 (GII+GIC+GII-ADE)
Titulación en la que se imparte:	Grado en Ingeniería Informática (GII) Ingeniería de Computadores (GIC) Ingeniería Informática y Administración y Dirección de Empresas (GII-ADE)
Departamento y Área de Conocimiento:	Ciencias de la Computación Ciencias de la Computación
Carácter:	Obligatoria (GII+GIC+GII-ADE)
Créditos ECTS:	6.0
Curso y cuatrimestre:	2º Curso - 2º Cuatrimestre (GII+GIC) 3º Curso - 2º Cuatrimestre (GII-ADE)
Profesorado:	José María Gutiérrez Martínez, Antonio García Cabot, Eva García López, José Manuel Lanza Gutiérrez, Sergio Caro Álvaro
Horario de Tutoría:	Consultar al comienzo de la asignatura
Idioma en el que se imparte:	Español

1a. PRESENTACIÓN

La asignatura Programación Avanzada introduce nuevos conceptos y técnicas de programación que permitirán a los estudiantes disponer de los conocimientos teóricos y el manejo de nuevas técnicas con los que abordar la realización de aplicaciones avanzadas y complejas en las que entren a formar parte aspectos avanzados de concurrencia, distribución y movilidad.

Prerrequisitos y Recomendaciones:

Se recomienda haber cursado la asignatura Programación.

1b. COURSE SUMMARY

This course introduces new concepts and programming techniques that allow students to handle the creation of advanced applications that use concurrency, distribution and Mobile.

To reach this goals, the course will show the origins, evolution, present time and future of the tools, languages and theories behind concurrency.

To follow this course is mandatory to have competences of basic and object oriented programming.

Java programming language is used along the course to show examples and to implement every type of program. Knowledge on this language is necessary to follow the course.

2. COMPETENCIAS

Competencias básicas, generales y transversales.

Esta asignatura contribuye a adquirir las siguientes competencias básicas, generales y transversales:

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 el apartado 5, anexo 2, de la resolución BOE-A-2009-12977.

CG6 - Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5, anexo 2, de la resolución BOE-A-2009-12977.

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.

TRU1 - Capacidad de análisis y síntesis.

TRU2 - Comunicación oral y escrita.

TRU3 - Capacidad de gestión de la información.

TRU4 - Capacidad de aprendizaje autónomo.

TRU5 - Capacidad para trabajar en equipo.

Competencias Específicas

Esta asignatura proporciona la(s) siguiente(s) competencia(s) específica(s):

CI8 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

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.

Resultados de aprendizaje

Los resultados de aprendizaje esperados, determinados a partir de las competencias específicas incluidas en la memoria verificada de la titulación como competencias específicas, son los siguientes:

RA1. Evaluar cómo afectan las capacidades y limitaciones de los lenguajes de programación más comunes a la creación de sistemas informáticos.

RA2. Describir la evolución de los lenguajes de programación, los diferentes paradigmas disponibles hoy día y sus principales características.

RA3. Analizar críticamente la evolución de los lenguajes de programación, los diferentes paradigmas disponibles hoy día y sus principales características en relación a su efecto sobre las decisiones de diseño de sistemas de información.

RA4. Aplicar la programación distribuida, concurrente y paralela, conociendo sus algoritmos fundamentales y las ventajas e inconvenientes de cada paradigma, para diseñar sistemas más eficaces o eficientes desde el punto de vista de rendimiento, eficiencia para el usuario u otros.

3. CONTENIDOS

Tema 1. Paradigmas de programación: Historia de los lenguajes de programación, Breve presentación de los distintos paradigmas. Comparación entre paradigmas, su utilidad y su influencia sobre los lenguajes de programación.

Tema 2. Introducción a la concurrencia: Evolución histórica. Terminología. Problemas de la concurrencia. Arquitecturas que soportan la concurrencia. Variables compartidas. Memoria distribuida. Hipótesis de justicia. Propiedades de seguridad y de vitalidad.

Tema 3. Concurrencia en memoria común: Procesos que cooperan y procesos que compiten. Ejemplos. Algoritmos. Espera activa. Cerrojos (Locks) y condiciones.

Tema 4. Mecanismos para la exclusión mutua: Semáforos, Regiones Críticas, RC Condicionales y Monitores.

Tema 5. Herramientas avanzadas de comunicación sincronizada: La especificación JSR166. Concurrencia y utilidades.

Tema 6. Control de hilos, tareas y pools: Tareas y Pools de Hilos.

Tema 7. Programación concurrente distribuida: paso de mensajes y sockets.

Tema 8. Programación concurrente distribuida: RPC y RMI.

Bloques de contenido	Total de clases, créditos u horas
Paradigmas de programación e iniciación a la programación concurrente. Temas 1, 2 y 3	2 ECTS
Mecanismos avanzados de comunicación sincronizada y gestión de tareas y pools. Tema 4, 5 y 6	2,5 ECTS
Programación concurrente distribuida. Temas 7 y 8	1,5 ECTS

4. METODOLOGÍAS DE ENSEÑANZA APRENDIZAJE. ACTIVIDADES FORMATIVAS

4.1. Distribución de créditos (especificar en horas)

Número de horas presenciales:	60 horas (56 horas de clase presencial + 4 horas de evaluación)
Número de horas del trabajo propio del estudiante:	90 (Incluye horas de estudio, elaboración de actividades, preparación de exámenes)
Total horas	150

4.2. Estrategias metodológicas, materiales y recursos didácticos

En el proceso de enseñanza-aprendizaje de los contenidos anteriormente reseñados se emplearán las siguientes actividades formativas:

- Clases Teóricas presenciales.
- Clases Prácticas: resolución de problemas presenciales.
- Prácticas en Laboratorio presenciales.

- Tutorías: individuales y/o grupales.

Además, en función de la naturaleza de las distintas partes de la materia objeto de estudio, se podrán utilizar, entre otras, las siguientes actividades formativas:

- Elaboración de trabajos con responsabilidad individual, pero con gestión de la información como equipo.
- Puesta en común de la información, problemas y dudas que aparezcan en la realización de los trabajos.
- Organización y realización de jornadas públicas con presentaciones orales y discusión de resultados.
- Utilización de Plataforma de Aula Virtual como elemento fundamental de comunicación de todas las actividades y materiales docentes de la asignatura.

Actividades presenciales:

- En el aula: exposición y discusión de los conocimientos básicos de la asignatura. Planteamiento y resolución teórica de ejercicios y supuestos relacionados. Orientadas a la enseñanza de las competencias específicas de la asignatura, especialmente las relacionadas con los conocimientos básicos y las técnicas de programación imperativa.
- En el laboratorio: planteamiento y desarrollo de ejercicios prácticos que permitan solventar problemas y analizar hipótesis y contribuyan al desarrollo de la capacidad de análisis de resultados, razonamiento crítico y comprensión de los métodos de resolución planteados. Servirán como base para la adquisición de las competencias genéricas descritas en el apartado 2.

Actividades no presenciales:

- Análisis y asimilación de los contenidos de la materia, resolución de problemas, consulta bibliográfica, preparación de trabajos individuales y grupales, realización de exámenes presenciales y autoevaluaciones. Orientadas especialmente al desarrollo de métodos para la autoorganización y planificación del trabajo individual y en equipo.
- Tutorías: asesoramiento individual y en grupos durante el proceso de enseñanza-aprendizaje, bien en forma presencial o a distancia.

Materiales y recursos:

- Bibliografía de referencia sobre la asignatura.
- Ordenadores personales.
- Entornos de desarrollo y manuales de uso de los mismos.
- Conexión a Internet.
- Plataforma de Aula Virtual y manuales de uso de las mismas.
- Proyector.

5. EVALUACIÓN: Procedimientos, criterios de evaluación y calificación

Preferentemente se ofrecerá a los alumnos un sistema de evaluación continua que tenga características de evaluación formativa de manera que sirva de realimentación en el proceso de enseñanza-aprendizaje por parte del alumno.

5.1. PROCEDIMIENTOS

La evaluación debe estar inspirada en los criterios de evaluación continua (Normativa de Evaluación de los Aprendizajes, NEA, art 3). No obstante, respetando la normativa de la Universidad de Alcalá se pone a disposición del alumno un proceso alternativo de evaluación final de acuerdo a la [Normativa de Evaluación de los Aprendizajes](#) según lo indicado en su Artículo 10, los alumnos tendrán un plazo de quince días desde el inicio del curso para solicitar por escrito al Director de la Escuela Politécnica

Superior su intención de acogerse al modelo de evaluación no continua aduciendo las razones que estimen convenientes. La evaluación del proceso de aprendizaje de todos los alumnos que no cursen solicitud al respecto o vean denegada la misma se realizará, por defecto, de acuerdo al modelo de evaluación continua. El estudiante dispone de dos convocatorias para superar la asignatura, una ordinaria y otra extraordinaria.

Convocatoria ordinaria

Evaluación continua:

El sistema de evaluación continua que se propone, se compone de: un ejercicio de laboratorio, un ejercicio de programación en laboratorio y pruebas escritas de carácter teórico/ práctico relativas a los conocimientos impartidos que se podrán realizar o entregar en el aula/laboratorio o a través de la Plataforma de Aula Virtual según el caso.

Información sobre valoración de las pruebas de evaluación continua:

- Cada una de las PEI de teoría tiene como objetivo evaluar las competencias de la parte de la materia sobre la que versará cada prueba, que debería haber adquirido el estudiante mediante el seguimiento de las clases presenciales, el estudio del material didáctico elaborado por los profesores, la lectura del material adicional propuesto para cada tema, la búsqueda de material adicional y la realización de ejercicios ya sean propuestos por los profesores u obtenidos por los estudiantes.

En cuanto a la Prueba de laboratorio, es obligatoria su realización, y consistirá en la realización de una aplicación completa en la que será preciso aplicar todos los conocimientos y destrezas adquiridos durante el curso. Así mismo, habrá un examen o defensa oral relativo a la realización de esta prueba.

El calendario de las PEI se dará a conocer a los estudiantes en los primeros días del curso a través de la "Agenda del Curso" que figurará en el Aula Virtual y donde se recogerán todas las incidencias de calendario que se pudiesen producir a lo largo del curso.

Prueba de Laboratorio	% sobre el total de la asignatura
PL	40

PEIs Teoría	% sobre el total de la asignatura
PEI1	15
PEI2	30
PEI3	15

Es requisito indispensable para superar la asignatura, haberse presentado y superado satisfactoriamente la evaluación de las competencias relacionadas con el conjunto de las pruebas teóricas PEI1, PEI2 y PEI3. Se entenderá que un alumno adquiere satisfactoriamente estas competencias, si su calificación en el conjunto de las pruebas relacionadas (de forma ponderada) es igual o superior al 40% de la nota máxima obtenible.

La superación de las prácticas (de laboratorio y de ejercicios/problemas) es requisito necesario para la superación de la asignatura. Se establecerán mecanismos adecuados para garantizar la viabilidad de la superación de las prácticas y de las actividades de la asignatura a aquellos alumnos que deban optar por la evaluación final o por la convocatoria extraordinaria.

Para superar la asignatura se debe obtener una calificación final ponderada de todas las pruebas de evaluación continua definidas igual o superior a 5 sobre 10 puntos.

Evaluación final:

El examen de evaluación final sólo es una opción para determinados supuestos de la normativa y debe ser solicitado por los estudiantes que cumplan los requisitos para ello y concedido por la dirección de la Escuela, de acuerdo con la normativa vigente de la Universidad de Alcalá.

Los estudiantes que se acojan a esta evaluación final, deberán realizar un examen de conocimientos teóricos de toda la materia de la asignatura y aportar todos los trabajos de laboratorio que hayan sido propuestos en la prueba de evaluación del laboratorio.

Al igual que en la evaluación continua, en la prueba teórica, que evalúa las competencias establecidas anteriormente, debe alcanzarse una calificación del 40% para superar la asignatura. Además, el alumno ha de superar las prácticas para poder superar la asignatura. En todo caso, la calificación final ponderada deberá ser igual o superior a 5 sobre 10 puntos.

Convocatoria extraordinaria

El procedimiento será el mismo que el descrito para la evaluación final en la convocatoria ordinaria, con los mismos tipos de entregables y ejercicios pero con enunciados diferentes.

5.2. EVALUACIÓN

CRITERIOS DE EVALUACIÓN

Se establecen los siguientes criterios de evaluación específicos para la asignatura:

- CE1.** El alumno conoce la historia y las características de los lenguajes de programación y es capaz de describir las diferencias entre los distintos paradigmas de programación.
- CE2.** El alumno comprende la programación concurrente, los diferentes tipos existentes y su evolución a partir de los modelos de programación secuencial.
- CE3.** El alumno conoce los principios en los que se basa el diseño de programas concurrente.
- CE4.** El alumno ha adquirido los conocimientos básicos sobre la coordinación de programas concurrentes.
- CE5.** El alumno ha adquirido los conocimientos avanzados del diseño de aplicaciones concurrentes de memoria común.
- CE6.** El alumno comprende la necesidad del uso de la programación concurrente distribuida y sus diferencias con la programación concurrente de memoria común.
- CE7.** El alumno sabe programar en un lenguaje que soporta concurrencia distribuida mediante sockets o RPC.
- CE8.** El alumno sabe desarrollar un sistema concurrente de memoria común utilizando cualquiera de los mecanismos disponibles en los lenguajes de programación.
- CE9.** El alumno ha adquirido los conocimientos básicos sobre tipos avanzados de programación concurrente.

En las siguientes tablas se indica el porcentaje en la calificación (entre 0 y 100) de cada prueba, y su relación con los criterios de evaluación, resultados de aprendizaje y competencias generales.

CRITERIOS DE CALIFICACIÓN

En la convocatoria **ordinaria–evaluación continua** la relación entre las competencias, resultados del aprendizaje, criterios e instrumentos de evaluación, es la siguiente.

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CG4, CI8	RA1, RA2, RA3	CE1, CE2	PEI1	15%
CG4, CG6, CI14	RA1, RA3, RA4	CE3, CE4, CE5, CE8, CE9	PEI2	30%
CG4, CG6, CI11, CI14	RA4	CE6, CE7	PEI3	15%
CG4, CG6, CI11, CI14	RA1, RA3, RA4	CE3, CE5, CE6, CE7, CE8, CE9	PL	40%

En la convocatoria **ordinaria–evaluación final** la relación entre las competencias, resultados del aprendizaje, criterios e instrumentos de evaluación, es la siguiente.

Competencia	Resultado de Aprendizaje	Criterio de Evaluación	Instrumento de evaluación	Peso en la calificación
CG4, CG6, CI8, CI11, CI14	RA1, RA2, RA3, RA4	CE1, CE2, CE3, CE4, CE5, CE6, CE7, CE8, CE9	PT	60%
CG4, CG6, CI8, CI11, CI14	RA1, RA3, RA4	CE3, CE5, CE6, CE7, CE8, CE9	PP	40%

Convocatoria extraordinaria

En el caso de la convocatoria extraordinaria se mantendrán los mismos porcentajes que se han establecido en el caso de la evaluación mediante ordinaria-evaluación final.

6. BIBLIOGRAFÍA

6.1. Bibliografía básica

- APUNTES DE LA ASIGNATURA, disponibles a través de la plataforma virtual.
- "Java Concurrency in practice". Goetz, Brian / Peierls, Tim / Bloch, Joshua / Bowbeer, Joseph / Holmes, David / Lea, Doug. 2007. Addison Wesley
- "Java in Distributed Systems". Boger, Marko. 2001. Wiley

6.2. Bibliografía complementaria

- "Programación concurrente". Palma Martínez, José Tomás / Garrido Carrera, M^a del Carmen / Sánchez Figueroa, Fernando / Quesada Arencibia, Alexis. 2003. Thomson
- "Programación concurrente en Java". Lea, Doug. 2000. Addison Wesley
- "Principles of Concurrent and Distributed Programming, Second Edition". M. Ben-Ari. Addison-Wesley. 2006.
- "The Art of Concurrency". Clay Breshears. O'Reilly Media, Inc. 2009.

- "Concurrency: State Models and Java Programs". Jeff Magee & Jeff Kramer. John Wiley & Sons. 2006. Programación Distribuida
- "Sistemas Distribuidos. Conceptos y diseño". Coulouris, George / Dollimore, Jean / Kindberg, Tim. 2001. Addison Wesley
- "Aplicaciones Distribuidas en Java con RMI". Caballé, Santi / Xhafa, Fatos. 2008. Delta Publicaciones
- "Introduccion a la Programacion Paralela". Almeida, F. / Gimenez, D. / Mantas, Jose Miguel / Vidal, A.M. Paraninfo Cengage Learning. 2008.
- "Concurrent Systems. Operating Systems, Database and Distributed Systems". Bacon, J. 1998. Addison Wesley
- "Object-oriented Reuse, Concurrency and Distribution". Atkinson, Colin. 1991. Addison Wesley
- "Parallel Program Design". Chandy, Mani / Misra, Jayedev. 1996. Addison Wesley
- "The Java Programming Language". Arnold, Ken / Gosling, James / Holmes, David. 2005. Addison Wesley
- "Concurrent and Real-Time Programming in Java". Wellings, Andrew. 2004. John Wiley & Sons
- "Concurrent Programming Concepts". Brinch-Hansen, P. 1973. ACM Computing Survey
- "Java Distributed Computing". Farley, J. 1998. O'Reilly & Associates, Sebastopol

NOTA INFORMATIVA

Durante el desarrollo de las pruebas de evaluación han de seguirse las pautas marcadas en el Reglamento por el que se establecen las Normas de Convivencia de la Universidad de Alcalá, así como las posibles implicaciones de las irregularidades cometidas durante dichas pruebas, incluyendo las consecuencias por cometer fraude académico según el Reglamento de Régimen Disciplinario del Estudiantado de la Universidad de Alcalá.