



Universidad
de Alcalá

GUÍA DOCENTE

Programación Avanzada

Grado en Ingeniería Informática
Grado en Ingeniería de Computadores
Universidad de Alcalá

Curso Académico 2019/2020
Curso 2º – Cuatrimestre 2º

GUÍA DOCENTE

Nombre de la asignatura:	Programación Avanzada
Código:	780014
Titulación en la que se imparte:	Grado en Ingeniería Informática Grado en Ingeniería de Computadores
Departamento:	Departamento Ciencias de la Computación
Carácter:	Obligatoria
Créditos ECTS:	6
Curso y cuatrimestre:	Segundo Curso / Segundo Cuatrimestre
Profesorado:	José María Gutiérrez Martínez, Antonio García Cabot, Eva García López, Ana Castillo Martínez
Horario de Tutoría:	El horario de Tutorías se indicará el primer día de clase.
Idioma en el que se imparte:	Español e Inglés

1.a 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:

Es obligatorio haber cursado la asignatura Programación

1.b 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 these 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 generales:

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 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 de la resolución BOE-A-2009-12977.

Competencias específicas:

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 (se pueden especificar los temas si se considera necesario)	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:	56 horas + 4 horas de exámenes
Número de horas del trabajo propio del estudiante:	90 horas
Total horas	150 horas

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 de calificación

El sistema de calificación se ajustará a la “NORMATIVA REGULADORA DE LOS PROCESOS DE EVALUACIÓN DE LOS APRENDIZAJES”, aprobada en Consejo de Gobierno de la Universidad de Alcalá de 24 de marzo de 2011.

La evaluación de la adquisición de competencias tendrá en cuenta la actitud y el interés del alumno. Los estudiantes tendrán la opción preferente de evaluación continua mediante las Pruebas de Evaluación Continua (PEC) distribuidas a lo largo del cuatrimestre. Además, los estudiantes podrán solicitar optar a una Evaluación Final si lo justifican en tiempo y forma.

Evaluación Continua:

El sistema de evaluación continua que se propone, se compone de: entregas periódicas de ejercicios de laboratorio, un ejercicio final 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 PEC de teoría (T1, T2 y T3) 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 las PEC de práctica, es obligatoria la realización de todas ellas:

- Las PEC L1 y L2 consistirán en la entrega de un trabajo de resolución práctica de uno o varios problemas de programación. Tendrán un plazo fijo de entrega.
- La PEC práctica L3 consistirá en la realización de una aplicación completa en la que será preciso aplicar todos los conocimientos y destrezas adquiridas durante el curso.
- La PEC práctica L4 consistirá en la realización en el laboratorio o aula designado de modificaciones propuestas sobre la PEC L3 entregada o sobre otro elemento de software o la explicación de la implementación realizada de ciertos aspectos de la PEC L3.

El calendario de las PEC 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.

Valoración de las Pruebas de Evaluación Continua (PEC) sobre la nota total:

La nota final se calculará aplicando un 60% sobre la nota de teoría y un 40% sobre la nota de laboratorio, siendo el peso de cada prueba:

PECs Laboratorio	% sobre el total de la asignatura
L1	2,5
L2	2,5
L3	20
L4	15

PECs Teoría	% sobre el total de la asignatura
T1	20
T2	25
T3	15

En ningún caso, la no superación de una PEC finalizará el proceso de evaluación continua.

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 sumativa, 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 las PECs L1, L2 y L3, realizar la prueba equivalente a las PECs de teoría y la PEC L4.

Evaluación Extraordinaria:

Durante el mes de Junio se llevará a cabo una nueva evaluación para los estudiantes que no hayan superado la anterior, tanto en su modalidad continua como de prueba final. Consistirá en una prueba igual a la de la evaluación final, con los mismos tipos de entregables y ejercicios pero con enunciados diferentes.

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. La nomenclatura utilizada para los instrumentos de evaluación es la siguiente, las PEC de teoría se denominan PEI más el número de la prueba y las PEC de laboratorio PL más el número de la prueba.

Convocatoria ordinaria:

Competencias	Resultado Aprendizaje	Criterio de Evaluación	Instrumento de Evaluación	Porcentaje
CG4, CI8, CI14	RA1, RA2, RA3	CE1, CE2, CE3, CE4	PEI1	20
CG4, CG6, CI14	RA1, RA3, RA4	CE5, CE8, CE9	PEI2	25
CG4, CG6, CI11, CI14	RA4	CE6, CE7	PEI3	15
CG6, CI14	RA4	CE3	PL1	2,5
CG6, CI11,	RA4	CE6, CE7	PL2	2,5

CI14				
CG4, CG6, CI11, CI14	RA1, RA3, RA4	CE5, CE6, CE7, CE8, CE9	PL3	20
CG4, CG6, CI11, CI14	RA1, RA3, RA4	CE7, CE8, CE9	PL4	15

Convocatoria final y extraordinaria: En esta convocatoria se realizará una prueba teórica (PT), prueba práctica (PP) y examen de laboratorio (EL).

Competencias	Resultado Aprendizaje	Criterio de Evaluación	Instrumento de Evaluación	Porcentaje
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	25
CG4, CG6, CI8, CI11, CI14	RA1, RA3, RA4	CE7, CE8, CE9	EL	15

BIBLIOGRAFÍA

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

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
- "Introducción a la Programación Paralela". Almeida, F. / Gimenez, D. / Mantas, José 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, Jaydev. 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