

GUÍA DOCENTE

PROGRAMACION AVANZADA

GRADO EN INGENIERÍA DE LA CIBERSEGURIDAD

CURSO 2026-27



I.-Identificación de la Asignatura

Tipo	OBLIGATORIA
Período de impartición	2 curso, 1Q semestre
Nº de créditos	6
Idioma en el que se imparte	Castellano

II.-Presentación

El objetivo de la asignatura es presentar los **fundamentos de la programación y el diseño orientados a objetos**. La asignatura aborda la enseñanza de la orientación a objetos tanto desde el punto de vista del diseño (usando **UML**), como desde el de la implementación (usando **Java**). Al final de la asignatura, el alumnado deberá ser capaz de:

- Conocer y comprender los conceptos básicos de programación orientada a objetos.
- Conocer y comprender los elementos esenciales del lenguaje Java
- Conocer y comprender conceptos básicos de diseño de programas orientados a objetos.
- Diseñar aplicaciones usando el paradigma de la programación orientada a objetos y el lenguaje UML.
- Codificar programas usando el paradigma de la programación orientada a objetos en el lenguaje Java.

Como **requisito previo** para cursar esta asignatura se necesitarán los conocimientos y competencias relacionados con la asignatura **Introducción a la Programación** detallados más adelante.

Se recomienda al alumno haber superado con éxito la asignatura **Estructuras de Datos**.

II.B-Presentación en inglés

The objective of the course is to present the fundamentals of object-oriented programming and design. The course approaches the teaching of object-oriented programming from both the design point of view (using UML) and the implementation point of view (using Java). At the end of the course, students should be able to: Know and understand the basic concepts of object-oriented programming. Know and understand the essential elements of the Java language. Know and understand basic concepts of object-oriented program design. Design applications using the object-oriented programming paradigm and the UML language. To code programs using the object-oriented programming paradigm in the Java language. As a prerequisite to take this course, the knowledge and competences related to the course Introduction to Programming detailed below will be required. It is recommended that the student has successfully passed the Data Structures course.

III.-Resultados de Aprendizaje



CG4. Capacidad para dirigir y liderar las actividades objeto de los proyectos del ámbito de la informática y la ciberseguridad comprendiendo los criterios de calidad que rigen dichas actividades investigadora y profesional.

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.

CG13. Capacidad para concebir, desarrollar, implantar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería como instrumento para el aseguramiento de su calidad.

CG15. Capacidad para aplicar conocimientos a su trabajo o vocación de una forma profesional. Capacidad para elaborar y defender argumentos y resolver problemas dentro de su área de estudio.

CG18. Capacidad para aplicar las habilidades de aprendizaje adquiridas necesarias para emprender estudios posteriores con un alto grado de autonomía.

CE06. Conocer y aplicar los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos

CE07. Comprender, diseñar y utilizar de forma eficiente los tipos y estructuras de datos más adecuados para la resolución de un problema.

- Leyenda para las titulaciones no adaptadas al RD 822/21: **CB** - competencias básicas, **CG** - competencias generales, **CE** - competencias específicas, **CT** - competencias transversales.
- Leyenda para las titulaciones adaptadas al RD 822/21: **CON** - conocimientos, **COM** - competencias, **HAB** - habilidades.



IV.-Contenido

IV.A.-Temario de la asignatura

Capítulo 1 - Introducción a la programación orientada a objetos

- 1.1. Complejidad del software: origen y tratamiento, Herramientas para tratar la complejidad, Evolución de los lenguajes de programación, Paradigmas de programación
- 1.2. Programación orientada a objetos: Abstraer, Encapsular, Jerarquizar, Modularizar, Tipo, Concurrencia, Persistencia

Capítulo 2 - Introducción al lenguaje Java

- 2.1. Introducción: Breve historia de Java, La máquina virtual de Java, Compiladores de Java, Entornos de desarrollo para Java, El programa hola mundo, Los comentarios, Juego de instrucciones
- 2.2. Expresiones básicas y control de flujo: Tipos primitivos, Conversiones entre tipos primitivos, Operadores, Control del flujo
- 2.3. Las clases: Las propiedades en detalle, Los métodos en detalle, Creación de objetos, Destrucción de objetos, Bloques de inicialización, Los modificadores de control de acceso a los miembros en detalle, Los modificadores de uso de los miembros en detalle, Modificadores en la definición de clases generales, Clases internas, Definición de arrays de objetos y de tipos primitivos
- 2.4. Los paquetes

Capítulo 3 - Herencia y genericidad en POO

- 3.1. La herencia de clases: Herencia en clases internas, Redefinición de miembros, La herencia y los constructores
- 3.2. Polimorfismo dinámico: El casting de referencias a objetos
- 3.3. Las interfaces

Capítulo 4 - El paquete java.lang

- 4.1. La clase Object: La clase ClassLoader
- 4.2. Clases para el manejo de cadenas de caracteres
- 4.3. Las excepciones: Tipos de excepciones, El uso práctico de las excepciones
- 4.4. Sobrecarga de los métodos básicos de object: Interfaz Cloneable, Los métodos equals y hashCode, Interfaz Comparable
- 4.5. La concurrencia: La palabra reservada synchronized, Mecanismos de compartición y comunicación
- 4.6. Los tipos enumerados
- 4.7. Envolturas

Capítulo 5 - Diseño de Clases

- 5.1. Diseño top-down apoyado en UML: Relaciones de dependencia entre objetos., Relaciones de asociación entre objetos, Relaciones entre clases2
- 5.2. Diseño Dirigido por Pruebas: Pruebas automáticas, Principios del Diseño Dirigido por Pruebas, Objetivos y ventajas del enfoque TDD

Capítulo 6 - Genericidad y estructuras de datos predefinidas

- 6.1. La genericidad: Clases e interfaces genéricas, Métodos con tipos parametrizados
- 6.2. Collection, El EnumSet
- 6.3. Map
- 6.4. Iteradores
- 6.5. La clase Collections
- 6.6. El resto del paquete java.util

Capítulo 7 - Entrada salida en POO

- 7.1. Streams de bytes
- 7.2. Streams de caracteres: Streams del sistema, StreamTokenizer
- 7.3. Manejo de ficheros
- 7.4. La interfaz Serializable

Capítulo 8 - Interfaces de usuario

- 8.1 Componentes del GUI de Java
- 8.2 Listeners
- 8.3 Containers
- 8.4 Layouts



IV.B.-Actividades formativas	
Tipo	Descripción
Resolución de ejercicios, problemas, casos	Ejercicios prácticos sobre programación
Prácticas	Prácticas de Programación
Tutorías académicas	Pruebas de evaluación continua



V.-Tiempo de trabajo del alumnado (30h grado y 25h máster)

Clases teóricas	14
Clases de resolución de ejercicios, problemas, casos, etc.	14
Prácticas en laboratorios experimentales, tecnológicos, clínicos, campo, etc.	24
Realización de pruebas	8
Tutorías académicas	40
Actividades relacionadas: jornadas, seminarios, etc.	8
Preparación de clases teóricas	20
Preparación de prácticas/ejercicios/casos	40
Preparación de pruebas	12
Total de horas de trabajo del alumnado	180

VI.-Metodología y plan de trabajo

Tipo	Periodo	Contenido
Prácticas	Semana 1 a Semana 15	MD3 y MD6 - Simultáneamente a cada uno de los temas se realizan clases de problemas y casos prácticos, con y sin ordenador
Pruebas	Semana 6 a Semana 8	MD8 - Evaluación de los temas 1, 2 y 3. (Examen tipo test 1)
Pruebas	Semana 12 a Semana 14	MD8 - Evaluación de los temas 4, 5 y 6. (Examen tipo test 2)
Prácticas	Semana 10 a Semana 13	MD4 y MD5 - Los alumnos realizan en grupo fuera de clase una práctica de diseño. De dicha práctica se realiza una primera reunión de explicación de requisitos, a la semana se hace una reunión de seguimiento, una semana más tarde los alumnos entregan la práctica y finalmente, la siguiente semana se realiza una sesión de revisión del resultado correcto.

Prácticas	Semana 13 a Semana 15	MD4 y MD5 - Cada alumno realiza fuera de clase una práctica de implementación y la entrega en el aula virtual. La práctica se explica en una primera reunión con los alumnos. Los alumnos puede solicitar tutorías respecto al desarrollo de dicha práctica. Dicha práctica se corrige junto con el examen de programación.
Prácticas	Semana 10 a Semana 10	MD1 y MD2 - Durante el curso se imparten lecciones magistrales de los Temas 1 a Tema 7. Dichas clase se acompañan de diapositivas y un libro de texto escrito por los profesores

VII.-Método de evaluación

El modelo de evaluación general es la evaluación continua, tal como establece el Reglamento de evaluación de los resultados de aprendizaje de la Universidad Rey Juan Carlos.

Deberán utilizarse todos los sistemas de evaluación establecidos para la asignatura en la memoria de la titulación, excepto aquellos que tuviesen una ponderación mínima del 0%, que podrán utilizarse en los cursos académicos en los que el profesorado lo considere oportuno. Cada uno de los sistemas de evaluación podrá ser aplicado mediante una o más actividades de evaluación, coherentes con ese sistema. Ninguna de las actividades de evaluación podrá superar individualmente el 60% de la calificación global de la asignatura.

La suma de las actividades de evaluación no revaluables no podrá superar el 40% de la calificación global de la asignatura y, en general, no deberían tener nota mínima (salvo en el caso de actividades de carácter práctico en las que, estrictamente, no pudieran reproducirse en la convocatoria extraordinaria las condiciones de evaluación de la convocatoria ordinaria).

El alumnado que no consiga superar la asignatura en la convocatoria ordinaria, o no se hayan presentado, podrán presentarse a la convocatoria extraordinaria únicamente a las actividades de evaluación revaluables no superadas.

El uso de herramientas basadas en inteligencia artificial está prohibido en cualquier ejercicio y prueba de evaluación de la asignatura salvo que el profesor lo indique de forma explícita en el enunciado o en el material de esta. En caso de que esté permitido, se darán indicaciones precisas de qué usos están permitidos y cuáles no.

La distribución y características de las actividades de evaluación son las que se describen a continuación.

VII.A.- Descripción de las actividades de evaluación y su ponderación

Sistema de evaluación	Revaluable en	Ponderación	Actividad de evaluación	Nota mínima	Contenidos	Fecha
SE.1 - Pruebas escritas de respuesta abiertas o tipo test	Sí (mismo formato que en convocatoria ordinaria)	50%	Examen tipo Test (10%)	1/10	Temas 1, 2 y 3	Semanas 6 a 8
			Examen tipo Test (10%)	1/10	Temas 4, 5 y 6	Semanas 11 a 13
			Examen de pregunta abierta (30%)	5/10	Diseño orientado a objetos (todos los temas)	Fecha oficial de convocatoria ordinaria
SE.2 - Resolución de problemas y casos prácticos	No*	20%	Trabajo en grupo de diseño en UML	-	Diseño orientado a objetos (temas 1 y 5)	Entrega en la semana 12 o 13
SE.3 - Prácticas con ordenador	Sí (mismo formato que en convocatoria ordinaria)	30%	Examen de pregunta abierta de programación basada en la práctica **	5/10	Programación orientada a objetos (todos los temas)	Fecha oficial de convocatoria ordinaria

* La no revaluabilidad de la parte de resolución de problemas y casos prácticos, que corresponde a un 20% de la nota, se deriva de que dicha práctica implica una carga de trabajo continuada e importante por parte del alumno (varias semanas de desarrollo) con seguimiento y apoyo por parte del profesorado (exposición de requisitos, seguimiento del diseño y exposición del nuevo diseño). Es por ello que no es posible repetirla en un periodo posterior a la convocatoria ordinaria. Los estudiantes con dispensa académica deberá asistir al menos a dos tutorías: una donde se les explicará la práctica y otra donde se les hará un seguimiento de la misma.

** El alumno desarrollará en Java de manera individual una aplicación original fuera del tiempo de clase contando para ello de un periodo de tiempo amplio y suficiente. Dicha aplicación estará sujeta a unos requisitos de funcionamiento, de estilo del código y de diseño previamente establecidos. Las desviaciones a peor leves respecto a dichos requisitos supondrán una penalización en la nota final, las desviaciones grandes a peor supondrán el suspenso de la práctica. Cualquier copia, total o parcial, de código (de una IA, de una web, de un compañero...) también será motivo de suspenso. Finalmente, el día del examen de la convocatoria ordinaria o extraordinaria, se le propondrá al alumno la realización de ciertas modificaciones a la aplicación por él desarrollada. El éxito o fracaso en la realización de dichas modificaciones, junto a las penalizaciones acumuladas, constituirán la nota de la prueba práctica de programación.

Cálculo de la nota final

- La nota final se calcula como la media ponderada de las notas de las pruebas evaluables según los porcentajes indicados, siempre y cuando se hayan superado con la nota mínima indicada para cada una de ellas.
- Para aprobar la asignatura hay que obtener al menos un 5 en la media ponderada.
- Si alguna de las pruebas evaluables no se ha superado con la nota mínima necesaria para hacer media, la nota final de la asignatura será el mínimo entre la media ponderada de las pruebas que haya realizado y 4.
- Si no se ha presentado a alguna de las pruebas evaluables en la convocatoria ordinaria que requiera de nota mínima, la nota final será el mínimo entre la media ponderada de las pruebas que haya realizado y 4. Además, de cara al cómputo de la nota, las pruebas no realizadas serán evaluadas con la nota de 0.
- Si no se ha presentado a ninguna prueba evaluable, la nota final será "No presentado".

Convocatoria ordinaria

El cálculo de la nota seguirá las pautas descritas en "Cálculo de la nota final".

Todas las pruebas revaluables realizadas durante el periodo docente que no hayan superado la **nota mínima** podrán ser recuperadas en la fecha oficial de la convocatoria ordinaria.

Convocatoria extraordinaria

El cálculo de la nota seguirá las mismas pautas que en la convocatoria ordinaria.

En la convocatoria extraordinaria todas las pruebas revaluables se realizarán en la fecha oficial de la convocatoria. Los estudiantes que hayan superado la nota mínima en cualquiera de las pruebas revaluables no podrán volver a presentarse a dicha prueba.

De manera excepcional si un estudiante, que habiendo superado la nota mínima en todas las pruebas revaluables no hubiesen logrado la calificación de aprobado, podrá presentarse a la prueba reevaluable con menor nota obtenida en la convocatoria ordinaria.

—

Convocatoria adelantada de septiembre

El cálculo de la nota seguirá las mismas pautas que en la convocatoria ordinaria.

En la convocatoria adelantada de septiembre, se realizarán todas las pruebas de evaluación, revaluables y no revaluables, en la fecha oficial de la convocatoria.

—

Conducta académica

El uso de la IA generativa está prohibido en cualquier ejercicio y prueba de evaluación de la asignatura.

En el caso de **fraude académico** en alguna actividad de evaluación, se otorgará una calificación de cero puntos en dicha actividad lo que, para aquellas actividades con nota mínima superior a cero implica el suspenso en la convocatoria correspondiente.

Se recuerda además que, atendiendo al artículo 9.d) de la **Normativa de Convivencia de la Universidad Rey Juan Carlos** (<https://www.urjc.es/images/Universidad/Presentacion/normativa/normativa%20convivencia%20universitaria.pdf>) el **fraude académico** en alguna actividad de evaluación se considera **falta grave**. Las sanciones correspondientes a las faltas graves, según el artículo 11 de la referida normativa, son la expulsión temporal de la Universidad, o la pérdida al derecho de la convocatoria ordinaria.

VII.B.- Evaluación del alumnado con dispensa académica de asistencia a clase

La concesión de Dispensa Académica de Asistencia a Clase (DAAC) no implica que se quede automáticamente eximido de participar en las actividades de evaluación continua ni en las actividades formativas presenciales de asistencia obligatoria establecidas en la guía docente. Una vez concedida la dispensa, el alumnado deberá contactar con el docente, que podría proponerle las adaptaciones que considere convenientes, siempre que garanticen la adquisición y adecuada evaluación de los resultados de aprendizaje previstos. El alumnado deberá mantener a lo largo de curso una comunicación fluida con el docente para que este le proporcione información sobre las fechas en que se realizarán esas actividades formativas y de evaluación, en caso de que su programación no estuviese ya fijada y a disposición del alumnado en el momento de la concesión de la dispensa.

Asignatura con posibilidad de dispensa: Sí

VII.C.- Revisión y reclamación de las actividades de evaluación

Se realizará conforme al Reglamento de evaluación de los resultados de aprendizaje de la Universidad Rey Juan Carlos.

VII.D.- Alumnado con discapacidad o necesidades educativas especiales

Con el fin de garantizar la igualdad de oportunidades, la no discriminación, la accesibilidad universal y la mayor garantía de éxito académico, las adaptaciones curriculares para alumnado con discapacidad o con necesidades educativas especiales serán pautadas por la Unidad de Atención a Personas con Discapacidad, de acuerdo con la normativa que regula el servicio de Atención a Estudiantes con Discapacidad, aprobada por Consejo de Gobierno de la Universidad Rey Juan Carlos.

Para que esas adaptaciones puedan realizarse, será requisito la emisión de un informe de adaptaciones curriculares por parte de dicha Unidad, por lo que el alumnado con discapacidad o necesidades educativas especiales deberán contactar con este servicio para analizar conjuntamente las distintas opciones.

VII.E.- Conducta académica, integridad y honestidad académica



La Universidad Rey Juan Carlos está plenamente comprometida con los más altos estándares de integridad y honestidad académica, por lo que estudiar en la URJC supone asumir y suscribir los valores de integridad y la honestidad académica recogidos en el Código Ético de la Universidad.

Para acompañar este proceso, la Universidad dispone de la Normativa de Convivencia de la Universidad Rey Juan Carlos y de diferentes herramientas (antiplagio, supervisión) que ofrecen una garantía colectiva para el completo desarrollo de estos valores esenciales.

Las herramientas basadas en inteligencia artificial pueden mejorar la experiencia de aprendizaje, pero debe utilizarse de forma ética, legal, honesta y transparente. Por este motivo, deben seguirse las recomendaciones de uso adecuado indicadas por los docentes de la asignatura.

VIII.-Recursos y materiales didácticos

Bibliografía básica

Aprende a programar con Java, 2ª edición. Alfonso Jiménez Marín, Franciso Manuel Pérez Montes. Ediciones Paraninfo, S.A (11 de marzo de 2016)

Programación Orientada a Objetos en JAVA. Francisco Blasco. Ra-Ma. Edición: 1 (3 de diciembre de 2021)

Programación orientada a objetos con java usando BlueJ. David Barnes, Michael Kolling. Pearson Alhambra. Edición: 1 (15 de junio de 2017)

Effective Java (Third Edition): A Programming Language Guide (Java Series). Joshua Bloch. Addison-Wesley Professional; Edición: 3. (27 de diciembre de 2017)

JAVA 8 (Programación). Rogers Cadenhead. Anaya; Edición: edición (23 de octubre de 2014)

Java 8 in Action: Lambdas, Streams, and functional-style programming. Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft. Manning Publications; Edición: 1 (28 de agosto de 2014)

Análisis y diseño en sistemas orientados a objetos con UML, 3a Ed. Simon Bennett, Steve Mcrobb, Ray Farmer. McGraw-Hill Interamericana de España S.L., 2007

Bibliografía complementaria

Introducción a los patrones de diseño: Un enfoque práctico. Oscar J Blancarte Iturralde. CreateSpace Independent Publishing Platform, Edición: 1 (18 de octubre de 2016)

Java SE8 for the Really Impatient: A Short Course on the Basics. Cay S. Horstmann. Addison-Wesley Educational Publishers Inc; Edición: 01 (24 de enero de 2007)

97 Things Every Java Programmer Should Know. Kevlin Henney. O'Reilly Media, Inc, USA (1 de noviembre de 2019)

Curso de Desarrollo Web: HTML, CSS y JavaScript. Edición 2018 (Manuales Imprescindibles). Mario Rubiales Gómez. ANAYA MULTIMEDIA; Edición: edición (12 de octubre de 2017)

UML. Aplicaciones En Java Y C++. Carlos Jiménez de Parga. Ra-Ma; Edición: 1 (23 de febrero de 2015)

IX.-Profesorado

Nombre y apellidos	DAVID ORTEGA DEL CAMPO
Correo electrónico	david.ortega.delcampo@urjc.es
Departamento	Informática y Estadística
Campus de impartición	Móstoles
Categoría	Profesor/a Permanente Laboral
Titulación académica	Doctor
Responsable de asignatura	Si
Horario de tutorías	Para consultar las tutorías póngase en contacto con el/la profesor/a a través de correo electrónico
Nº de Quinquenios	1
Nº de Sexenios	0
Nº de Sexenios de transferencia	0
Nº de evaluaciones positivas Docencia	2



Nombre y apellidos	JOSÉ FRANCISCO VÉLEZ SERRANO
Correo electrónico	jose.velez@urjc.es
Departamento	Informática y Estadística
Campus de impartición	Móstoles
Categoría	Catedrático/a de Universidad
Titulación académica	Doctor
Responsable de asignatura	No
Horario de tutorías	Para consultar las tutorías póngase en contacto con el/la profesor/a a través de correo electrónico
Nº de Quinquenios	4
Nº de Sexenios	3
Nº de Sexenios de transferencia	1
Nº de evaluaciones positivas Docencia	6
Nombre y apellidos	MARÍA DEL SOTO MONTALVO HERRANZ
Correo electrónico	soto.montalvo@urjc.es
Departamento	Informática y Estadística
Campus de impartición	Móstoles
Categoría	Titular de Universidad
Titulación académica	Doctor
Responsable de asignatura	No
Horario de tutorías	Para consultar las tutorías póngase en contacto con el/la profesor/a a través de correo electrónico
Nº de Quinquenios	4
Nº de Sexenios	2
Nº de Sexenios de transferencia	0
Nº de evaluaciones positivas Docencia	6

