

GUÍA DOCENTE DISEÑO SOFTWARE

GRADO EN INGENIERÍA DE ROBÓTICA SOFTWARE

CURSO 2026-27

Fecha de publicación: 07-07-2026



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
En esta asignatura el estudiante aprenderá a analizar, diseñar, implementar y probar software usando el paradigma de Orientación a Objetos, que es el más utilizado actualmente en desarrollo de software. El estudiante será capaz de emplear métodos, técnicas y herramientas ingenieriles para la construcción de aplicaciones robustas y mantenibles mediante el uso de este paradigma. Se estudiará el lenguaje de programación C++, que es uno de lenguajes más usados en programación orientada a objetos (POO). Por otro lado, se usará un lenguaje de modelado universal, como es UML, para modelar el diseño orientado a objetos (DOO). Concretamente, usaremos los diagramas de clases de UML para representar de forma visual las clases y sus relaciones, así como los diagramas de actividad de UML para demostrar el flujo de control del sistema. También se introducirán los patrones de diseño, que ayudarán a construir un software orientado a objetos flexible y fácilmente mantenible, además de facilitar un desarrollo software de mejor calidad con menos tiempo y esfuerzo. Recomendaciones Antes de realizar esta asignatura el alumno debiera haberse familiarizado con los conceptos de programación adquiridos en la asignatura Fundamentos de la Programación, así como los conocimientos relacionados con algoritmos y estructuras de datos adquiridos en la asignatura Algoritmos y Estructuras de Datos.
II.B-Presentación en inglés
In this course, the student will learn to analyze, design, implement, and test software using the Object-Oriented paradigm, which is the most widely used in software development today. The student will be able to employ engineering methods, techniques, and tools for building robust and maintainable applications using this paradigm. C++, which is one of the most widely used languages in Object-Oriented Programming (OOP), will be studied. Additionally, a universal modeling language, such as UML, will be used to model Object-Oriented Design (OOD). Specifically, we will use UML class diagrams to visually represent classes and their relationships, as well as UML activity diagrams to demonstrate the system's control flow. Design patterns will also be introduced, which will help in building flexible and easily maintainable Object-Oriented software, and will facilitate better quality software development with less time and effort. Recommendations: Before taking this course, students should be familiar with the programming concepts acquired in the Fundamentals of Programming course, as well as the knowledge related to algorithms and data structures acquired in the Algorithms and Data Structures course.



III.-Resultados de Aprendizaje

CG1. Conocimiento de materias básicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías, así como que le dote de una gran versatilidad para adaptarse a nuevas situaciones.

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

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

CE7. Capacidad de analizar, diseñar, programar y usar estructuras de datos, bases de datos y abstracciones de alto nivel eficientes y adecuadas a la resolución de problemas en el ámbito de la Ingeniería Robótica.

- 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

1. Introducción a la POO
2. Clases. Generalidades
3. Clases. Uso avanzado
4. Sobrecarga de operadores
5. Herencia
6. Polimorfismo
7. Plantillas
8. Depuración y Excepciones
9. Entrada/Salida de flujos
10. Ficheros
11. Strings
12. Estructuras de datos y librería STL
13. Patrones de diseño

IV.B.-Actividades formativas

Tipo	Descripción
Asistencia a clases teóricas	Ejercicios prácticos (programas) sincronizados con el temario con periodicidad de una semana dentro del laboratorio.
Lecturas	Lecturas de la referencias recomendadas.
Asistencia a clases teóricas	Creacion de un sistema completo que incorpore los distintos bloques temáticos de la asignatura.

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

Clases teóricas	15
Clases de resolución de ejercicios, problemas, casos, etc.	15
Prácticas en laboratorios experimentales, tecnológicos, clínicos, campo, etc.	15
Realización de pruebas	15
Tutorías académicas	48
Actividades relacionadas: jornadas, seminarios, etc.	0
Preparación de clases teóricas	15
Preparación de prácticas/ejercicios/casos	30
Preparación de pruebas	27
Total de horas de trabajo del alumnado	180

VI.-Metodología y plan de trabajo

Tipo	Periodo	Contenido
Clases Teóricas	Semana 1 a Semana 13	Colaboración en la resolución de dudas de otros compañeros de curso e implicación en las discusiones técnicas surgidas en el foro del Aula Virtual.
Clases Teóricas	Semana 1 a Semana 13	Tema correspondiente según temario.
Prácticas	Semana 2 a Semana 13	Proyectos prácticos de la asignatura.
Laboratorios	Semana 2 a Semana 15	Ejercicio práctico de programación del tema.
Tutorías académicas	Semana 1 a Semana 13	Tutorías académicas para resolución de dudas.
Pruebas	Semana 13 a Semana 20	Examen ordinario y extraordinario, que se realizarán en las fechas oficiales publicadas por la universidad. Los detalles de estos exámenes se encuentran en el siguiente apartado, "Evaluación".



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

Evaluación ordinaria

La evaluación para la convocatoria ordinaria consistirá en:

- 1.- Entregas periódicas de ejercicios prácticos de programación.
- 2.- Examen teórico y examen práctico sobre los contenidos impartidos durante el curso.

La ponderación de cada prueba y tarea se detalla en la siguiente tabla:

Ítem	Peso	Nota mínima	Fecha	Tipo	Contenido
Ejercicios prácticos	40%	4 sobre 10	A lo largo del curso	Reevaluable en convocatoria extraordinaria	Contenido impartido hasta la práctica
Examen teórico	30%	4 sobre 10	Fecha oficial de examen	Reevaluable en convocatoria extraordinaria	Todo el contenido del curso
Examen práctico	30%	4 sobre 10	Fecha oficial de examen	Reevaluable en convocatoria extraordinaria	Todo el contenido del curso

Para aprobar la asignatura es necesario superar las notas mínimas en todos los elementos de evaluación y que la nota media ponderada sea igual o superior a 5.

Evaluación extraordinaria

Los alumnos que no consigan superar la evaluación ordinaria, o no se hayan presentado, serán objeto de la realización de una evaluación extraordinaria para verificar la adquisición de las competencias establecidas en la guía.

El mecanismo de esta evaluación es similar a la evaluación ordinaria, descrita en el apartado anterior. Salvo que, para la entrega de prácticas, la fecha máxima será el día del examen oficial de convocatoria extraordinaria.

Evaluación adelantada

Habrà un examen teórico (40%) y un examen práctico (60%). La nota mínima para el examen teórico es un 4 y la nota mínima para el examen práctico es un 4.



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

Andrei Alexandrescu. Modern C++ Design: Generic Programming and Design Patterns Applied. Addison-Wesley.

Nicolai M. Josuttis. Object-Oriented Programming in C++. John Wiley & Sons.

Carlos Jiménez de Parga. UML. Arquitectura de aplicaciones en Java, C++ y Python. RA-MA S.A

Ambler, S. The Object Primer: Agile Model-Driven Development with UML 2.0, Third Edition. Nueva York: Cambridge University Press, 2005.

Bibliografía complementaria

Arlow, J. e I. Neustadt. UML and the Unied Process: Practical Object-Oriented Analysis and Design, Second Edition. Boston: Addison-Wesley Professional, 2006.

Rumbaugh, J., I. Jacobson y G. Booch. The Unied Modeling Language User Guide, Second Edition. Boston: Addison-Wesley Professional, 2006.

IX.-Profesorado

Nombre y apellidos	DAVID ROLDÁN ÁLVAREZ
Correo electrónico	david.rolدان@urjc.es
Departamento	Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación
Campus de impartición	Fuenlabrada
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	1
Nº de Sexenios	2
Nº de Sexenios de transferencia	0
Nº de evaluaciones positivas Docencia	2
Nombre y apellidos	JULIO MANUEL VEGA PÉREZ
Correo electrónico	julio.vega@urjc.es
Departamento	Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación
Campus de impartición	Fuenlabrada



Categoría	Profesor/a Contratado/a Doctor/a
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	1
Nº de Sexenios de transferencia	0
Nº de evaluaciones positivas Docencia	3