



# Guía Didáctica - GRADO

## ASIGNATURA: Proyectos de Programación

Título: Grado en Ingeniería Informática

Módulo: Programación de Computadores

Créditos: 6 ECTS

Código: 21GIIN

# Índice

1. Organización general.....	3
1.1. Datos de la asignatura.....	3
1.2. Introducción a la asignatura.....	3
1.3. Competencias y resultados de aprendizaje .....	4
2. Contenidos/temario .....	7
<b>4. Metodologías Docentes .....</b>	<b>9</b>
5. Evaluación .....	10
5.1. Sistema de evaluación.....	10
5.2. Sistema de Calificación.....	11
6. Bibliografía .....	12

# 1. Organización general

## 1.1. Datos de la asignatura

<b>MÓDULO</b>	Programación de Computadores
<b>ASIGNATURA</b>	21GIIN - Proyectos de Programación 6 ECTS
<b>Carácter</b>	Obligatorio
<b>Curso</b>	Tercero
<b>Cuatrimestre</b>	Primero
<b>Idioma en que se imparte</b>	Castellano
<b>Requisitos previos</b>	Se recomienda haber cursado las asignaturas de Estructuras de datos y algoritmos, Bases de datos y Fundamentos de ingeniería de software
<b>Dedicación al estudio recomendada por ECTS</b>	25 horas

## 1.2. Introducción a la asignatura

*En esta asignatura integraremos algunos conceptos vistos anteriormente, sobre todo en las asignaturas de: Estructuras de datos y algoritmos, Bases de datos y Fundamentos de ingeniería de software.*

*Para poder identificar partes susceptibles de ser factorizadas en la especificación, diseño e implementación de un programa, para resolverlas una sola vez, y usar de manera efectiva los mecanismos de la orientación a objetos para hacer la factorización. Veremos los principios del diseño y la programación orientada a objetos, su justificación, y las ventajas e inconvenientes de adoptar este paradigma en un proyecto de programación. Aprenderemos como diseñar una interfaz de usuario razonablemente cómoda y eficaz para el programa y finalmente desarrollaremos un proyecto de programación de tamaño medio, en grupo, de acuerdo con un plan de desarrollo y una arquitectura proporcionados de antemano.*

### **1.3. Competencias y resultados de aprendizaje**

#### **COMPETENCIAS BÁSICAS**

**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**

**CG1** - Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

**CG2** - Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.

**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 la resolución.

**CG5** - Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.

**CG8** - 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.

**CG10** - Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.

## **COMPETENCIAS ESPECÍFICAS DE LA ASIGNATURA**

**R1** - Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

**R5** - Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.

**R6** - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

**R7** - Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

**R8** - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

**R11** - 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.

**R14** - Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

## **RESULTADOS DE APRENDIZAJE**

Al finalizar esta asignatura se espera que el estudiante sea capaz de:

**RA.1.** Identificar partes susceptibles de ser factorizadas en la especificación, diseño e implementación de un programa, para resolverlas una sola vez, y usar de manera efectiva los mecanismos de la orientación a objetos para hacer la factorización.

**RA.2.** Describir los principios del diseño y la programación orientada a objetos, su justificación, y las ventajas e inconvenientes de adoptar este paradigma en un proyecto de programación.

**RA.3.** Diseñar una interfaz de usuario razonablemente cómoda y eficaz para el programa.

**RA.4.** Desarrollar un proyecto de programación de tamaño medio, en grupo, de acuerdo con un plan de desarrollo y una arquitectura proporcionados de antemano.

## 2. Contenidos/temario

1. Conceptos de OO (programación orientada a objetos)
2. Especificación: diagramas de clases y casos de uso
3. Diseño: arquitectura en 3 capas
4. El lenguaje de programación Java
5. Depuración de programas
6. Documentación
7. Conceptos básicos de diseño de interfaces

## 3. Actividades Formativas

Bloque de 24 ECTS, Asignatura de 6 ECTS -> dividir por 4

ACTIVIDAD FORMATIVA	HORAS	PRESENCIALIDAD
Clases expositivas	15	60
Resolución de ejercicios prácticos	20	30
Prácticas de laboratorios virtuales	25	20
Tutorías	15	0
Trabajo Autónomo	75	0

Original ANECA (Bloque)

ACTIVIDAD FORMATIVA	HORAS	PRESENCIALIDAD
Clases expositivas	60	60
Resolución de ejercicios prácticos	80	30
Prácticas de laboratorios virtuales	100	20
Tutorías	60	0
Trabajo Autónomo	300	0



## 4. Metodologías Docentes

Clases teóricas impartidas como lecciones magistrales o exposiciones, en las que además de presentar el contenido de la asignatura, se explican los conceptos fundamentales y se desarrolla el contenido teórico.

Colección de tareas que el alumnado llevará a cabo a lo largo de toda la asignatura, entre las que podemos encontrar: análisis de casos, resolución de problemas, prácticas de laboratorios, comentarios críticos de textos, análisis de lecturas, etc.

Sesiones periódicas entre el profesorado y el alumnado para la resolución de dudas, orientación, supervisión, etc.

Trabajo tanto individual como grupal para la lectura crítica de la bibliografía, estudio sistemático de los temas, reflexión sobre problemas planteados, resolución de actividades propuestas, búsqueda, análisis y elaboración de información, investigación e indagación, así como trabajo colaborativo basado en principios constructivistas.

## 5. Evaluación

### 5.1. Sistema de evaluación

El Modelo de Evaluación de estudiantes en la Universidad se sustenta en los principios del Espacio Europeo de Educación Superior (EEES), y está adaptado a la estructura de formación virtual propia de esta Universidad. De este modo, se dirige a la evaluación de competencias.

Es requisito indispensable aprobar el portafolio y la prueba final con un mínimo de 5 para ponderar las calificaciones.

Sistema de Evaluación	Ponderación
<b>Portafolio*</b>	<b>60 %</b>
Realización de un proyecto en grupo donde se integraran y reforzaran varios conceptos vistos en asignaturas anteriores: programación orientada a objetos, programación estructurada, documentación, modelo en capas, uso de bases de datos para la persistencia, especificación formal, control de errores, y validación.	
Sistema de Evaluación	Ponderación
<b>Prueba final*</b>	<b>40 %</b>
Prueba con un pequeño componente teórico sobre los conceptos vistos en la asignatura y un fuerte componente practico con preguntas sobre el proyecto desarrollado en grupo.	

**\*Es requisito indispensable para superar la asignatura aprobar cada apartado (portafolio y prueba final).**

Atendiendo a la Normativa de Evaluación de la Universidad, se tendrá en cuenta que la utilización de **contenido de autoría ajena** al propio estudiante debe ser citada adecuadamente en los trabajos entregados. Los casos de plagio serán sancionados con suspenso (0) de la actividad en la que se detecte. Asimismo, el uso de **medios fraudulentos durante las pruebas de evaluación** implicará un suspenso (0) y podrá implicar la apertura de un expediente disciplinario.

## 5.2. Sistema de Calificación

La calificación de la asignatura se establecerá en los siguientes cálculos y términos:

Nivel de Competencia	Calificación Oficial	Etiqueta Oficial
Muy competente	9,0 - 10	Sobresaliente
Competente	7,0 - 8,9	Notable
Aceptable	5,0 - 6,9	Aprobado
Aún no competente	0,0 - 4,9	Suspense

El nivel de competencia en cada una de las actividades realizadas se medirá, teniendo en cuenta **criterios generales derivados de la consecución de los resultados de aprendizaje**, que en términos generales y en función de la adecuación en el planteamiento de los contenidos generales y contenidos específicos, valorarán por norma general y en trabajos escritos, la corrección de la estructura formal y organización del discurso (semántica, sintaxis y léxico) valorándose además la originalidad, creatividad y argumentación de las intervenciones utilizando referencias bibliográficas.

Sin detrimento de lo anterior, el alumnado dispondrá de una **rúbrica simplificada en el campus** que mostrará los aspectos que valorará el docente, como así también los **niveles de desempeño que tendrá en cuenta para calificar las actividades vinculadas a cada resultado de aprendizaje**.

## 6. Bibliografía

Moreno, Pérez, Juan. Programación orientada a objetos, RA-MA Editorial, 2015.

Flórez, Fernández, Héctor Arturo. Programación orientada a objetos usando java, Ecoe Ediciones, 2012.

García, Llinás, Luis Fernando. Todo lo básico que debería saber: sobre programación orientada a objetos en Java, Ediciones de la U, 2010.

Teniente, L. E., Costal, C. D., & Sancho, S. M. R. (2003). Especificación de sistemas software en uml.

The UML diagrams(2009). (2nd ed. ed.). London: BCS Learning & Development Limited.

Kimmel, P. (2008). Manual de uml.