



# Guía Didáctica - GRADO

## ASIGNATURA: **Desarrollo del Software**

Título: **Grado en Ingeniería Informática**

Módulo: **Menciones**

Créditos: **6 ECTS**

Código: **40GIIN**

# Índice

1. Organización general.....	3
1.1. Datos de la asignatura.....	3
1.2. Introducción a la asignatura.....	3
1.4. Competencias y resultados de aprendizaje .....	3
2. Contenidos/temario .....	5
3. Evaluación .....	10
3.1. Sistema de evaluación.....	10
3.2. Sistema de Calificación.....	11
4. Bibliografía .....	12

# 1. Organización general

## 1.1. Datos de la asignatura

<b>MÓDULO</b>	Menciones
<b>MATERIA</b>	Mención en Ingeniería del Software
<b>ASIGNATURA</b>	Desarrollo del Software 6 ECTS
<b>Carácter</b>	Obligatorio
<b>Curso</b>	Tercero
<b>Cuatrimestre</b>	Segundo
<b>Idioma en que se imparte</b>	Castellano
<b>Requisitos previos</b>	No existen
<b>Dedicación al estudio recomendada por ECTS</b>	25 horas

## 1.2. Introducción a la asignatura

*A través de este curso se pretende hacer una introducción al entorno profesional del desarrollo de software que permita al alumno incorporarse a un proyecto conociendo de antemano el contexto en el que se desarrolla. Al final de este curso el alumno debe de haber asimilado y ser capaz de manejar los conceptos del desarrollo del software.*

## 1.3. Competencias y resultados de aprendizaje

### COMPETENCIAS GENERALES

CG.1.- Conocer y manejar los conceptos del desarrollo del software.

CG.2.- Aplicar los principios y métodos que son propios de proyectos de ingeniería a proyectos de software.

## **COMPETENCIAS ESPECÍFICAS DE LA ASIGNATURA**

C.E.1.- Aprender a describir el análisis y el diseño de un programa en el lenguaje unificado de modelado (UML).

C.E.2.- Entender la arquitectura de un programa de software y conocer los modelos de ciclo de vida que pueden seguirse en el desarrollo de un proyecto.

C.E.3.- Seleccionar y aplicar los patrones de arquitectura adecuados a tipo de proyecto.

## **RESULTADOS DE APRENDIZAJE**

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

RA.1.- Manejar los conceptos del desarrollo del software.

RA.2.- Seleccionar el modelo de ciclo de vida más apropiado para un proyecto de software.

RA.3.- Crear diagramas en UML.

RA.4.- Aplicar los patrones de diseño adecuados a un proyecto de software.

## 2. Contenidos/temario

### Unidad Competencial 1 / Tema 1: Introducción al desarrollo de software

- 1.1. Ingeniería del software
- 1.2. Objetivos del curso
- 1.3. El entorno del desarrollo de software profesional
- 1.4. Características del software
- 1.5. Proceso de creación de un producto de software

### Unidad Competencial 2 / Tema 2: Modelos de ciclo de vida

- 2.1. Modelo en cascada
- 2.2. Modelo en V
- 2.3. Modelo iterativo
- 2.4. Modelo Ágil
  - 2.4.1. Manifiesto por el Desarrollo Ágil de Software
  - 2.4.2. Principios del Manifiesto Ágil
  - 2.4.3. Scrum
  - 2.4.4. Kanban
  - 2.4.5. Modelo de "Programación Extrema".

### Unidad Competencial 3 / Tema 3: UML

- 3.1. Elementos de UML
  - 3.1.1. Elementos
  - 3.1.2. Elementos de comportamiento
  - 3.1.3. Elementos de agrupación
  - 3.1.4. Elementos de anotación
- 3.2. Relaciones en UML
- 3.3. Diagramas UML
  - 3.3.1. Diagramas de paquetes
  - 3.3.2. Diagramas de clases
  - 3.3.3. Diagramas de objetos
  - 3.3.4. Diagramas de estructuras compuestas
  - 3.3.5. Diagramas de componentes
  - 3.3.6. Diagramas de despliegue
  - 3.3.7. Diagramas de casos de uso

- 3.3.8. Diagramas de actividad
- 3.3.9. Diagramas de máquina de estados (o diagrama de estados)
- 3.3.10. Diagramas de comunicación (o de colaboración)
- 3.3.11. Diagramas de secuencia
- 3.3.12. Diagramas de tiempos
- 3.3.13. Diagramas de interacción

#### Unidad Competencial 4 / Tema 4: Diseño y arquitectura

- 4.1. Definición de arquitectura del software
- 4.2. Patrones de arquitectura de software
  - 4.2.1. Descomposición modular
  - 4.2.2. Orientada a servicios
  - 4.2.3. Pipe-and-filter
  - 4.2.4. Pizarra (o Blackboard)
  - 4.2.5. Cliente-servidor
  - 4.2.6. Arquitectura de tres niveles
  - 4.2.7. Basado en eventos
- 4.3. Diseño de software
  - 4.3.1. Fases del diseño de software

#### Unidad Competencial 5 / Tema 5: Pruebas o Tests

- 5.1. Casos de test
- 5.2. Tipos de pruebas
  - 5.2.1. Pruebas estáticas y dinámicas
  - 5.2.2. Pruebas incrementales
  - 5.2.3. Pruebas contra especificación
  - 5.2.4. Pruebas manuales y automáticas
  - 5.2.5. Pruebas de caja negra y caja blanca
  - 5.2.6. Pruebas funcionales
  - 5.2.7. Pruebas unitarias
  - 5.2.8. Pruebas de integración
  - 5.2.9. Pruebas de sistema
  - 5.2.10. Pruebas de aceptación
  - 5.2.11. Pruebas no funcionales
- 5.3. Herramientas

#### Unidad Competencial 6 / Tema 6: Entrega, Mantenimiento y Evolución

- 6.1. Entrega
  - 6.1.1. Planes de entrega
  - 6.1.2. Entrega automática
  - 6.1.3. Retirada

## 6.2. Mantenimiento

### 6.2.1. Ciclo de mantenimiento

### 6.2.2. Refactorización

### 3. Actividades Formativas

<b>ACTIVIDAD FORMATIVA</b>	<b>HORAS</b>	<b>PRESENCIALIDAD</b>
Clases expositivas	120	60
Resolución de ejercicios prácticos	160	30
Prácticas de laboratorios virtuales	200	20
Tutorías	120	0
Trabajo Autónomo	600	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>50 %</b>
Colección de tareas realizadas por el alumnado y establecidas por el profesorado. La mayoría de las tareas aquí recopiladas son el resultado del trabajo realizado dirigido por el profesorado en las actividades, tutorías, etc. Esto permite evaluar, además de las competencias conceptuales, otras de carácter más práctico, procedimental o actitudinal.	
Sistema de Evaluación	Ponderación
<b>Prueba final*</b>	<b>50 %</b>
La realización de una prueba cuyas características son definidas en cada caso por el correspondiente profesorado.	

**\*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	Suspenso

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** 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

Anderson, D. J. (2010). *Kanban: successful evolutionary change for your technology business*. Blue Hole Press. [ISBN 0-9845214-0-2](#)

IEEE Standard for System and Software Verification and Validation, in IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004), vol., no., pp.1-223, 25 May 2012  
doi: 10.1109/IEEESTD.2012.6204026  
<https://ieeexplore.ieee.org/document/6204026>

Garlan, David, and Mary Shaw (Enero, 1994). *An Introduction to Software Architecture*. Carnegie Mellon University Technical Report, CMU-CS-94-166.

Martin, R. C. (2002). *Agile software development: principles, patterns, and practices*. Prentice Hall.