

GUIA DOCENTE DE LA ASIGNATURA (∞)
DESARROLLO DE SOFTWARE

Curso 2019-2020
(Fecha última actualización: 21/05/2019)
(Fecha de aprobación en Consejo de Departamento: 27/05/2019)

MÓDULO	MATERIA	CURSO	SEMESTRE	CRÉDITOS	TIPO
Formación de Especialidad 2: Ingeniería del Software	Desarrollo y Gestión de Proyectos	3º	Sexto	6	Obligatoria
PROFESORES ⁽¹⁾			DIRECCIÓN COMPLETA DE CONTACTO PARA TUTORÍAS (Dirección postal, teléfono, correo electrónico, etc.)		
• María del Mar Abad Grau: grupos A, DS1, DS2, DS3 COORDINADOR DE LA ASIGNATURA: María del Mar Abad Grau			Dpto. Lenguajes y Sistemas Informáticos, 3ª planta, Despacho nº 25. Correo electrónico: mabad@ugr.es @ugr.es		
			HORARIO DE TUTORÍAS Y/O ENLACE A LA PÁGINA WEB DONDE PUEDAN CONSULTARSE LOS HORARIOS DE TUTORÍAS ⁽¹⁾		
			María del Mar Abad Grau: https://lsi.ugr.es/lsi/mabad		
GRADO EN EL QUE SE IMPARTE			OTROS GRADOS A LOS QUE SE PODRÍA OFERTAR		
Grado en Ingeniería Informática			Doble Grado en Ingeniería Informática y Matemáticas Doble Grado en Ingeniería Informática y Administración de Empresas Grado en Ingeniería de Tecnologías de Telecomunicación		

1 . Consulte posible actualización en Acceso Identificado > Aplicaciones > Ordenación Docente(∞)
Esta guía docente debe ser cumplimentada siguiendo la "Normativa de Evaluación y de Calificación de los estudiantes de la Universidad de Granada"
([http://secretariageneral.ugr.es/pages/normativa/fichasugr/ncg7121/!](http://secretariageneral.ugr.es/pages/normativa/fichasugr/ncg7121/))



PRERREQUISITOS Y/O RECOMENDACIONES (si procede)
Se recomienda la superación de los contenidos y adquisición de competencias de las materias de formación básica y de rama.
BREVE DESCRIPCIÓN DE CONTENIDOS (SEGÚN MEMORIA DE VERIFICACIÓN DEL GRADO)
Patrones de diseño. Clasificación y composición de patrones. Arquitecturas de software. Técnicas de desarrollo dirigidas por modelos. Modelado de negocio. Técnicas de verificación y validación de software. Pruebas unitarias. Mantenimiento y evolución del software.
COMPETENCIAS GENERALES Y ESPECÍFICAS
<p>Competencias Generales del Título</p> <ul style="list-style-type: none"> • E1. 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, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas. • E5. 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 <p>Competencias Específicas del módulo</p> <ul style="list-style-type: none"> • IS1. Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software. • IS3. Capacidad de dar solución a problemas de integración de componentes-software en función de las estrategias, estándares y tecnologías disponibles. • IS4. Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de técnicas de desarrollo de software dirigidas por modelos. <p>Competencias Transversales</p> <ul style="list-style-type: none"> • T3. Capacidad de comunicación oral y escrita en el ámbito académico y profesional, con especial énfasis en la redacción de documentación técnica.
OBJETIVOS (EXPRESADOS COMO RESULTADOS ESPERABLES DE LA ENSEÑANZA)
<ul style="list-style-type: none"> • Conocer los patrones que se pueden aplicar a un diseño y a su estructura y • Clasificación de los patrones y su importancia como herramienta para mejorar la calidad de un diseño. • Adquirir destreza en la identificación de los patrones aplicables a un determinado problema. • Comprender correctamente el concepto de componente-software y saber aplicarlo en le desarrollo • Conocer las técnicas de desarrollo dirigidas por modelos, sus ventajas e inconvenientes. • Conocer y saber aplicar diferentes niveles de prueba a un software • Introducirse en el modelado de procesos de negocio, así como en sus técnicas fundamentales y herramientas



- Conocer la importancia de un buen mantenimiento y evolución con respecto al ciclo de desarrollo
- Aprender a evaluar la calidad de un software de acuerdo con las características y estándares de calidad actuales

TEMARIO DETALLADO DE LA ASIGNATURA

TEMARIO TEÓRICO:

- **Tema 1. Desarrollo utilizando patrones-software.**
 1. Análisis y diseño basado en patrones
 2. Estudio del catálogo GoF de patrones diseño y arquitectónicos
 3. Resolución de ejercicios.
- **Tema 2. Arquitecturas de Software.**
 1. Estilos arquitectónicos
 2. Notaciones actuales para descripción arquitectónica
 3. Arquitecturas orientas a componentes y servicios
 4. Resolución de ejercicios de.
- **Tema 3. Desarrollo dirigido por modelos.**
 1. Fundamentos del desarrollo dirigido por modelos (MDD)
 2. Arquitecturas dirigidas por modelos (MDA) y factorías de software
 3. Modelado de negocio
 4. Arquitecturas de empresa/Sistemas de Información Empresarial
 5. Resolución de ejercicios.
- **Tema 4. Validación del software.**
 1. Objetivos y principios fundamentales.
 2. Validación y verificación.
 3. Relación con el ciclo de vida del software
 4. Técnicas, herramientas y marcos de trabajo para el desarrollo de pruebas del software
 5. Resolución de ejercicios.
- **Tema 5. Mantenimiento ye evolución del software**
 1. Principios y tipos de mantenimiento aplicables al software.
 2. El proceso de mantenimiento del software
 3. Actividades relacionadas con la correcta evolución del software
 4. Resolución de ejercicios.

TEMARIO PRÁCTICO:

Seminarios/Talleres

- **Seminario práctico 1:** Introducción al desarrollo de apps con Android
- **Seminario práctico 2:** Introducción a los servicios Web RESTful
- **Seminario práctico 3:** Introducción al desarrollo de pruebas unitarias en Java (JUnit)

Prácticas de Laboratorio

Práctica 1. Realizar la aplicación del catálogo GoF al diseño de varios programas que resuelven un conjunto de problemas propuestos. Analizar la adecuación del diseño e implementación obtenidos con respecto a los requisitos especificados para cada problema en el guión.

Práctica 2. Desarrollar una aplicación eligiendo los patrones de diseño más adecuadas a partir de una descripción informal de un problema. El problema consiste en la simulación de un sistema de control automático de conducción de un vehículo desarrollado con Java/Swing.



<p>Práctica 3. Realizar una propuesta de arquitectura software y aplicarla al desarrollo completo de una aplicación entre varias propuestas:</p> <ul style="list-style-type: none"> -App Android interactuando con un servidor REST desplegado en Internet -Aplicación Web RESTful receptiva y adaptable a diferentes dispositivos para visitar un museo <p>Práctica 4. Desarrollo de pruebas unitarias y de integración para evaluar la calidad de uno de los proyectos-software desarrollado en la práctica 1. Redacción de un informe de validación del software referido.</p>
BIBLIOGRAFÍA
<p>BIBLIOGRAFÍA FUNDAMENTAL:</p> <ul style="list-style-type: none"> • Bruegge, B. Dutoit, A.H. Ingeniería del software orientado a objetos. Pearson Education, 2002. • Larman, C. UML y patrones : una introducción al análisis y diseño orientado a objetos y al proceso unificado. Pearson Education, 2010 • Gamma, E., Helm, E., Johnson, R., Vlissides, J. Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995. Traducción al español de César Fernández Acebal y Juan Manuel Cueva Lovelle. Pearson Educación, 2009 <p>BIBLIOGRAFÍA COMPLEMENTARIA:</p> <ul style="list-style-type: none"> • Bruegge, B. Dutoit, A.H. Object-Oriented Software Engineering. Using UML Patterns, and Java. Pearson Education, 2014. • Fowler, M. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2014 • Fisher, M.S. Software Verification and Validation. Springer-Verlag, 2007 • Jazarbeck, S. Effective Software Maintenance and Evolution: A Reuse-Based Approach. Auerbach, 2007. • Rainsberger, J.B. JUnit Recipes: Practical Methods for Programmer Testing. Manning Publications, 2005.
ENLACES RECOMENDADOS
Al comienzo del curso se avisará de la plataforma Web y de las páginas auxiliares donde se encontrarán los enlaces recomendados para la asignatura.
METODOLOGÍA DOCENTE
<p>1. Lección magistral (clases teóricas-expositivas)(grupo grande)</p> <ul style="list-style-type: none"> • Descripción: Presentación en el aula de los conceptos propios de la materia haciendo uso de metodología expositiva con lecciones magistrales participativas y medios audiovisuales. Evaluación y examen de las capacidades adquiridas. • Propósito: Transmitir los contenidos de la materia motivando al alumnado a la reflexión, facilitándole el descubrimiento de las relaciones entre diversos conceptos y formarle una mentalidad crítica. Contenido en ECTS: 30 horas presenciales (1.2 ECTS). • Competencias: IS1, IS3, IS4, IS6, E1, E2, E3, E4, E5, E6, E7, E9, E10, E12, T1, T2, T4, T5, T6. • Metodología empleada: Lección Magistral, Resolución de Problemas, Resolución de Casos Prácticos, Exposición de Trabajos Tutelados. <p>2. Actividades prácticas (Clases prácticas de laboratorio)(grupo pequeño)</p> <ul style="list-style-type: none"> • Descripción: Actividades a través de las cuales se pretende mostrar al alumnado cómo debe actuar a partir de la aplicación de los conocimientos adquiridos.



- **Propósito:** Desarrollo en el alumnado de las habilidades instrumentales de la materia.
- **Contenido en ECTS:** 15 horas presenciales (0.6 ECTS).
- **Competencias:** IS1, IS3, IS4, IS6, E1, E2, E3, E4, E5, E6, E7, E9, E10, E12, T1, T2, T4, T5, T6.
- **Metodologías empleadas:** Taller de Programación, Resolución de Problemas, Aula de Informática, Desarrollo de Proyectos.

3. Seminarios (grupo pequeño)

- **Descripción:** Modalidad organizativa de los procesos de enseñanza y aprendizaje donde tratar en profundidad una temática relacionada con la materia. Incorpora actividades basadas en la indagación, el debate, la reflexión y el intercambio.
- **Propósito:** Desarrollo en el alumnado de las competencias cognitivas y procedimentales de la materia.
- **Contenido en ECTS:** 10 horas presenciales (0.4 ECTS).
- **Competencias:** IS1, IS3, IS4, IS6, E1, E2, E3, E4, E5, E6, E7, E9, E10, E12, T1, T2, T4, T5, T6.
- **Metodologías docentes:** Taller de Programación.

4. Actividades no-presenciales individuales (estudio y trabajo autónomo)

- **Descripción:** 1) Actividades (guiadas y no guiadas) propuestas por el profesor a través de las cuales y de forma individual se profundiza en aspectos concretos de la materia posibilitando al estudiante avanzar en la adquisición de determinados conocimientos y procedimientos de la materia; 2) Estudio individualizado de los contenidos de la materia; 3) Actividades evaluativas (informes, exámenes, ...).
- **Propósito:** Favorecer en el estudiante la capacidad para autorregular su aprendizaje, planificándolo, diseñándolo, evaluándolo y adecuándolo a sus especiales condiciones e intereses.
- **Contenido en ECTS:** 45 horas no presenciales (1.8 ECTS).
- **Competencias:** IS1, IS3, IS4, IS6, E1, E2, E3, E4, E5, E6, E7, E9, E10, E12, T1, T2, T4, T5, T6.

5. Actividades no presenciales grupales (Estudio y trabajo en grupo)

- **Descripción:** Actividades (guiadas y no guiadas) propuestas por el profesor a través de las cuales y de forma grupal se profundiza en aspectos concretos de la materia posibilitando a los estudiantes avanzar en la adquisición de determinados conocimientos y procedimientos de la materia.
- **Propósito:** Favorecer en los estudiantes la generación e intercambio de ideas, la identificación y análisis de diferentes puntos de vista sobre una temática, la generalización o transferencia de conocimiento y la valoración crítica del mismo.
- **Contenido en ECTS:** 45 horas no presenciales (1.8 ECTS).
- **Competencias:** IS1, IS3, IS4, IS6, E1, E2, E3, E4, E5, E6, E7, E9, E10, E12, T1, T2, T4, T5, T6, T8.

6. Tutorías académicas (grupo pequeño)

- **Descripción:** manera de organizar los procesos de enseñanza y aprendizaje que se basa en la interacción directa entre el estudiante y el profesor.
- **Propósito:** 1) Orientan el trabajo autónomo y grupal del alumnado; 2) profundizar en distintos aspectos de la materia y 3) orientar la formación académica-integral del estudiante.
- **Contenido en ECTS:** 5 horas presenciales, grupales e individuales (0.2 ECTS).
- **Competencias:** IS1, IS3, IS4, IS6, E1, E2, E3, E4, E5, E6, E7, E9, E10, E12, T1, T2, T4, T5, T6.

DEFINICIÓN DE GRUPO GRANDE Y PEQUEÑO:



- Los grupos grandes son los grupos de teoría o de prácticas
- Los grupos pequeños son grupos de 2 a 4 estudiantes

RÉGIMEN DE ASISTENCIA (EVALUACIÓN CONTINUA)

- La asistencia tanto a las clases prácticas como teóricas será obligatoria, siendo necesario superar una asistencia mínima del 70% de la clases de teoría y al menos 12 sesiones prácticas en la convocatoria ordinaria.
- La asistencia a las sesiones programadas para pruebas de evaluación y a los 3 seminarios propuestos en el temario práctico de la asignatura será obligatoria.

EVALUACIÓN (INSTRUMENTOS DE EVALUACIÓN, CRITERIOS DE EVALUACIÓN Y PORCENTAJE SOBRE LA CALIFICACIÓN FINAL, ETC.)

Todo lo relativo a la evaluación se regirá por la Normativa de evaluación y calificación de los estudiantes vigente en la Universidad de Granada, que puede consultarse en:

https://lsi.ugr.es/lsi/normativa_examenes

EVALUACIÓN CONTINUA

Preferentemente, la evaluación se ajustará al sistema de evaluación continua del aprendizaje del estudiante siguiendo el artículo 7 de la anterior Normativa.

En esta modalidad se utilizarán las siguientes técnicas de evaluación:

- Para la parte teórica se realizarán varias sesiones de evaluación (exámenes) y entregas de ejercicios sobre el desarrollo y los resultados de las actividades propuestas.
- Para la parte práctica se realizarán prácticas de laboratorio, resolución de problemas y desarrollo de proyectos, valorándose las entregas en fechas establecidas de los informes/memorias realizados para cada práctica por los alumnos, o en su caso las entrevistas personales con los alumnos y las sesiones de evaluación (exámenes). En concreto, sólo se utilizará el examen como método de evaluación en la práctica 1, examen que requerirá la implementación de código usando el ordenador. Las demás prácticas serán evaluadas según los informes/memorias presentados y mediante entrevistas personales.
- En los seminarios se valorará la resolución de los problemas y ejercicios que hayan sido resueltos y entregados en plazo por los alumnos, en su caso las entrevistas efectuadas durante el curso y la presentación de los trabajos asignados.
- Para los estudiantes que se acojan a la evaluación única final, las pruebas y la evaluación estarán regidas por los criterios que se exponen más adelante en este documento (ver apartado correspondiente)

La calificación global corresponderá a la puntuación obtenida de la ponderación de los diferentes aspectos y actividades que integran el sistema de evaluación. Por tanto, el resultado de la evaluación será una calificación numérica obtenida mediante la suma ponderada de las calificaciones obtenidas en la parte teórica, práctica y, en el caso de la evaluación continua, una parte relacionada con el trabajo autónomo de los alumnos, los seminarios impartidos y el aprendizaje basado en proyectos de desarrollo de software. La adaptación del sistema de evaluación general propuesto a las características de esta asignatura se ajustará a lo indicado en la siguiente tabla:



Actividades formativas	Ponderación
Parte teórica	40%
Parte práctica	50%
Otros (seminarios, ...)	10%

Para aprobar la asignatura es necesario obtener una calificación numérica superior o igual a 5 puntos (sobre 10). No obstante, además del requisito anterior, se establece adicionalmente que tanto la calificación correspondiente a la parte teórica como a la parte práctica sean mayores o iguales a 3,5 puntos (sobre 10).

El sistema de calificaciones se expresará mediante calificación numérica de acuerdo con lo establecido en el artículo 5 del R.D. 1125/2003 de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y validez en el territorio nacional.

CONVOCATORIA EXTRAORDINARIA

La evaluación de los estudiantes en la convocatoria extraordinaria se regirá por los mismos criterios y constará de las mismas pruebas que las indicadas en este documento para evaluación única final.

DESCRIPCIÓN DE LAS PRUEBAS QUE FORMARÁN PARTE DE LA EVALUACIÓN ÚNICA FINAL ESTABLECIDA EN LA "NORMATIVA DE EVALUACIÓN Y DE CALIFICACIÓN DE LOS ESTUDIANTES DE LA UNIVERSIDAD DE GRANADA"

La evaluación única final consistirá en dos pruebas de evaluación, una para la parte teórica y otra para la parte práctica, con las características que se indican a continuación:

- **Evaluación de teoría:** los estudiantes realizarán una única prueba escrita que constará de preguntas de teoría, problemas y ejercicios sobre la teoría impartida y resolución de un supuesto práctico
- **Evaluación de las prácticas:** los estudiantes realizarán una prueba en el laboratorio que consistirá en la realización con ordenador de un programa basado en un supuesto práctico similar a los realizados durante las prácticas regladas.

La ponderación de cada parte en la nota final será del 50% (parte teórica) y 50% (parte práctica). Para aprobar la asignatura se han de cumplir los siguientes requisitos:

La nota de la prueba de teoría ha de ser igual o superior al 40% del máximo de dicha prueba

La nota de la prueba de prácticas ha de ser igual o superior al 40% del máximo de dicha prueba

La suma ponderada de ambas notas ha de ser igual o superior al 50% del máximo posible que puede alcanzar dicha suma.

INFORMACIÓN ADICIONAL

