Facultad de Ciencias

GUÍA DOCENTE DE LA ASIGNATURA

G666 - Diseño de Software

Grado en Ingeniería Informática
Optativa. Curso 4

Curso Académico 2023-2024

1. DATOS IDENTIFICATIVOS

Título/s	Grado en Ingeniería Informática			Tipología y Curso	Optativa. Curso 4
Centro	Facultad de Ciencias				
Módulo / materia	MATERIA INGENIERÍA DEL SOFTWARE MENCIÓN EN INGENIERÍA DEL SOFTWARE				
Código y denominación	G666 - Diseño de Software				
Créditos ECTS	6	Cuatrimestre		Cuatrimestral (1)	
Web	http://moodle.unican.es/				
Idioma de impartición	Español	English friendly	Sí	Forma de impartición	Presencial

Departamento	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA
Profesor responsable	DIEGO GARCIA SAIZ
E-mail	diego.garcia@unican.es
Número despacho	Facultad de Ciencias. Planta: + 1. DESPACHO INVESTIGADOR (1068)
Otros profesores	PABLO SANCHEZ BARREIRO

2. CONOCIMIENTOS PREVIOS

Para poder superar esta asignatura, se considera imprescindible que el alumno posea unos sólidos conocimientos sobre programación básica, orientación a objetos, estructuras de datos y recursividad.

Además, el alumno deberá haber cursado las asignaturas de Ingeniería del Software I, Ingeniería del Software II y Bases de Datos. Como consecuencia de ello, el alumno deberá ser capaz de crear diagramas de clases UML, diseñar y ejecutar pruebas unitarias y conocer cómo se accede a una base de datos relacional desde código Java utilizando JDBC.

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS
Competencias Genéricas
(Conocimiento) 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.
(Aplicación) 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.
(Análisis) Reunir e interpretar datos relevantes (dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
(Comunicación) Transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
(Aprendizaje) Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
Capacidad de análisis, síntesis y evaluación.
Capacidad de gestión de la información.
Capacidad de resolución de problemas aplicando técnicas de ingeniería.
Capacidad para argumentar y justificar lógicamente las decisiones tomadas y las opiniones.
Aprendizaje autónomo.
Tener motivación por la calidad.
Competencias Específicas
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.
Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.
Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse.
Capacidad para diseñar soluciones apropiadas en uno o más dominios de aplicación utilizando métodos de la ingeniería del software que integren aspectos éticos, sociales, legales y económicos.
Competencias Básicas
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.
Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

3.1 RESULTADOS DE APRENDIZAJE
- Conocer las principales técnicas y criterios para analizar, diseñar y estructurar software.
- Aplicar las principales notaciones, estrategias y herramientas para el análisis y diseño de software.
- Utilizar patrones software para encontrar buenas soluciones a problemas de diseño y construcción del software.
- Aplicar las principales técnicas de modelado y diseño orientado a objetos, incluidos los aspectos avanzados de UML.

4. OBJETIVOS
El alumno entenderá y sabrá aplicar los principios GRASP y SOLID durante la elaboración de un diseño software.
El alumno entenderá y sabrá aplicar los principios del Diseño por Contrato.
El alumno entenderá el funcionamiento y sabrá aplicar un conjunto de patrones de diseño microarquitectónicos.
El alumno entenderá el funcionamiento y sabrá aplicar un conjunto de patrones de diseño arquitectónicos para el desarrollo de sistemas empresariales.
El alumno será capaz de evaluar las ventajas e incovenientes asociados a la aplicación de un determinado patrón de diseño.

5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES	
ACTIVIDADES	HORAS DE LA ASIGNATURA
ACTIVIDADES PRESENCIALES	
HORAS DE CLASE (A)	
- Teoría (TE)	20
- Prácticas en Aula (PA)	10
- Prácticas de Laboratorio Experimental(PLE)	
- Prácticas de Laboratorio en Ordenador (PLO)	30
- Prácticas Clínicas (CL)	
Subtotal horas de clase	60
ACTIVIDADES DE SEGUIMIENTO (B)	
- Tutorías (TU)	6
- Evaluación (EV)	8
Subtotal actividades de seguimiento	14
Total actividades presenciales (A+B)	74
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	
Trabajo autónomo (TA)	76
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
Total actividades no presenciales	76
HORAS TOTALES	150

6. ORGANIZACIÓN DOCENTE

CONTENIDOS		TE	PA	PLE	PLO	CL	TU	EV	TG	TA	TU-NP	EV-NP	Semana
1	Tema 1 - Principios Fundamentales del Diseño Software. Modularidad Software. Principios GRASP. Principios SOLID. Diseño por Contrato.	2,00	1,00	0,00	4,00	0,00	0,00	0,50	0,00	8,00	0,00	0,00	1-2
2	Tema 2 - Patrones de Diseño Software. Concepto de Patrón, Antipatrón y Refactorización. Patrones GoF. Otros Patrones no GoF. Inyección de Dependencias. Expresiones Lambda.	7,00	4,00	0,00	14,00	0,00	3,00	3,50	0,00	30,00	0,00	0,00	3-9
3	Tema 3 - Diseño de Arquitecturas Software Revisión del Concepto de Arquitectura. Vistas Arquitectónicas. Lenguajes de Descripción Arquitectónica. Evaluación de Arquitecturas Software. Patrones Arquitectónicos Comunes: Arquitecturas en Capas, Cliente-Servidor, Publicación-Suscripción, Tuberías y Filtros, Orientadas a Servicios, Microservicios.	2,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	4,00	0,00	0,00	9
4	Tema 4 - Arquitecturas de Sistemas de Información Empresarial Arquitecturas en Capas. Distribución de Capas en Arquitecturas Empresariales. Patrones para la Capa de Negocio. Domain-Driven Design. Patrones para la Capa de Persistencia. Patrones para la Capa de Servicio. Patrones para la Capa de Presentación.	7,00	5,00	0,00	10,00	0,00	3,00	4,00	0,00	30,00	0,00	0,00	10-15
5	Tema 5 - Introducción a Paradigmas de Diseño Software no Orientados a Objetos. Programación Funcional. Programación Orientada a Aspectos. Programación Reactiva.	2,00	0,00	0,00	2,00	0,00	0,00	0,00	0,00	4,00	0,00	0,00	1
TOTAL DE HORAS		20,00	10,00	0,00	30,00	0,00	6,00	8,00	0,00	76,00	0,00	0,00	

Esta organización tiene carácter orientativo.

TE	Horas de teoría
PA	Horas de prácticas en aula
PLE	Horas de prácticas de laboratorio experimental
PLO	Horas de prácticas de laboratorio en ordenador
CL	Horas de prácticas clínicas
TU	Horas de tutoría
EV	Horas de evaluación
TG	Horas de trabajo en grupo
TA	Horas de trabajo autónomo
TU-NP	Tutorías No Presenciales
EV-NP	Evaluación No Presencial

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Prueba Parcial Arquitecturas Empresariales	Evaluación en laboratorio	Sí	Sí	50,00

Calif. mínima	3,50
Duración	2 horas
Fecha realización	La dispuesta por el centro para la convocatoria ordinaria
Condiciones recuperación	
Observaciones	La prueba versará sobre los contenidos correspondientes al segundo bloque temático de la asignatura (Temas 3, 4 y 5), relativo al diseño de arquitecturas para Sistemas de Información Empresarial. La prueba consisitirá en una serie de ejercicios prácticos a realizar con la ayuda de un computador. Durante la prueba, el alumno deberá aplicar sus conocimientos sobre patrones de diseño para Sistemas de Información Empresarial a uno o más problemas concretos. Como parte de la prueba podrá ser necesaria la modificación de una o más de las prácticas realizadas durante las sesiones de prácticas con computadores. Por tanto, es responsabilidad del alumno acudir a esta prueba con dichas prácticas realizadas. Además, la prueba podrá contener una o más cuestiones de tipo teórico que exijan la elaboración de un argumento por parte del alumno. En ningún caso se harán preguntas teóricas de tipo memorístico. Aquellos alumnos que superen la calificación mínima exigida para esta prueba podrán considerar la misma como superada. Aquellos alumnos que no superen la calificación mínima exigida, o quieran mejorar dicha calificación, podrán repetir la prueba correspondiente al periodo de evaluación extraordinario.

Prueba Parcial Patrones de Diseño	Evaluación en laboratorio	No	Sí	50,00

Calif. mínima	3,50
Duración	2 horas
Fecha realización	Finales de octubre o principios de noviembre
Condiciones recuperación	
Observaciones	La prueba versará sobre los contenidos correspondientes al primer bloque temático de la asignatura (Temas 1 y Tema 2), relativos a los principios y patrones de diseño software. La prueba consisitirá en una serie de ejercicios prácticos a realizar con la ayuda de un computador. Durante la prueba, el alumno deberá aplicar sus conocimientos sobre patrones de diseño a uno o más problemas concretos. Como parte de la prueba, podrá ser necesaria la modificación de una o algunas de las prácticas realizadas durante las sesiones prácticas con computadores. Por tanto, es responsabilidad del alumno acudir a esta prueba con dichas prácticas realizadas. Además, la prueba podrá contener una o más cuestiones de tipo teórico que exijan la elaboración de un argumento por parte del alumno. En ningún caso se harán preguntas teóricas de tipo memorístico. Aquellos alumnos que superen la calificación mínima exigida para esta prueba podrán considerar la misma como superada. Aquellos alumnos que no superen la calificación mínima exigida o quieran mejorar dicha calificación, podrán recuperar la prueba en los periodos de evaluación ordinario y extraordinario.

TOTAL	100,00
Observaciones	

La calificación final de la asignatura estará determinada por la media ponderada de las diferentes pruebas parciales. Para ello, será condición necesaria superar las calificaciones mínimas exigidas en cada actividad. En caso de no superarse alguna calificación mínima, la calificación del alumno se calculará como el mínimo de 4,9 y la media ponderada obtenida.

Aquellas pruebas parciales en las cuales se supere dicha calificación mínima se considerarán como superadas para el resto del curso académico, no siendo necesaria su recuperación en las siguientes convocatorias, ya sean ordinarias o extraordinarias.

Con objeto de verificar la autoría de las soluciones proporcionadas por los alumnos y evitar plagios indeseados, el profesor podrá solicitar a los alumnos en cualquier momento pequeñas aclaraciones o modificaciones sobre el material entregado. En caso de que estas aclaraciones o modificaciones hagan sospechar de forma claramente fundada que el alumno no es el autor del correspondiente material, dicho alumno quedará automáticamente suspenso en la asignatura y el caso será expuesto a instancias superiores para que tomen las medidas disciplinarias que consideren oportunas.

Todas las pruebas evaluables serán pruebas prácticas a realizar mediante la utilización de computadores .
Para la realización de estas pruebas se dejará hacer uso de todo tipo de material impreso, incluyendo apuntes, ejercicios y libros.

Criterios de evaluación para estudiantes a tiempo parcial

Las únicas actividades estrictamente presenciales son la realización de las pruebas parciales y sus recuperaciones. Por tanto, no se requieren medidas especiales para los alumnos matriculados a tiempo parcial. Dado que todas las pruebas realizadas durante el cuatrimestre son recuperables en los periodos de evaluación tanto ordinarios como extraordinarios, estos alumnos podrán optar además por esquemas de evaluación donde el 100% de su calificación quede determinado por el resultado de las pruebas prácticas realizadas durante estos periodos de evaluación.

La fecha de realización de las diferentes pruebas parciales se anunciará durante las dos primeras semanas de curso, de manera que los alumnos a tiempo parcial puedan realizar con suficiente tiempo de antelación las gestiones oportunas para poder acudir a dichas pruebas.

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Marzo 2000.

Complementaria

Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3º Edición. Prentice Hall. Octubre 2004.

Robert C. Martin y Micah Martin. Agile Principles, Patterns and Practices in C#. Prentice Hall. Julio 2006.

Bertrand Meyer. Touch of Class: Learning to Program Well with Objects and Contracts. Springer. Septiembre 2009.

Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra. "Head First Design Patterns". O'Reilly. 2004

William J. Brown, Raphael C. Malveau, Hays W. "Skip" McCormick y Thomas J. Mowbray. AntiPatterns: Refactoring Software, Architectures and Projects in Crisis. Wiley. Abril 1998.

Richard Taylor, Nenad Medvidovic y Eric M. Dashofy. Software Architecture: Foundations, Theory, and Practice. Wiley. Enero 2009.

Felix Bachmann and Len Bass and Paul Clements and David Garlan and James Ivers and Reed Little and Robert Nord y Judith Stafford . Documenting Software Architectures: Views and Beyond. Addison-Wesley Professional. Octubre 2010.

Martin Fowler. "Patterns of Enterprise Application Architecture". Addison-Wesley Professional. Spetiembre 2002.

Eric Evans. "Domain-Driven Design". Addison-Wesley. Agosto 2003.

9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Magic Draw	Facultad de Ciencias			
Paquete Ofimático Básico	Facultad de Ciencias			
Visual Studio C#	Facultad de Ciencias			
Eclipse con Java Development Tools y Maven	Facultad de Ciencias			

10. COMPETENCIAS LINGÜÍSTICAS

- ☑ Comprensión escrita
- ☐ Comprensión oral
- ☐ Expresión escrita
- ☐ Expresión oral
- ☐ Asignatura íntegramente desarrollada en inglés

Observaciones

El alumno deberá hacer uso intensamente de numerosa bibliografía disponible exclusivamente en inglés.