Proyecto docente de la asignatura

Asignatura	DISEÑO DE SOFTWARE		
Materia	INGENIERIA DEL SOFTWARE		
Módulo	Tecnologías Específicas		
Titulación	Grado en INGENIERÍA INFORMÁTICA Mención Ingeniería del Software		
Plan	545	**Código**	46924
Periodo de impartición	2º Cuatrimestre	**Tipo/Carácter**	OB
Nivel/Ciclo	Grado	**Curso**	3
Créditos ECTS	6		
Lengua en que se imparte	Español		
Profesor/es responsable/s	Yania Crespo González-Carvajal		
Datos de contacto (E-mail, teléfono…)	yania[at]infor.uva.es		
Horario de tutorías	Véase www.uva.es → Grados → Grado en Ingeniería Informática → Tutorías , y buscar el nombre del profesor de la asignatura.		
Departamento	Informática		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Esta asignatura se encuentra situada en el tercer curso de Ingeniería Informática, junto a otras que conforman la materia “Ingeniería del Software” (ver apartado 1.2) necesaria para la mención del mismo nombre. La asignatura repasará el concepto de Proceso de Desarrollo para ubicar la etapa de Diseño en el proceso. Por otra parte, se profundizará en los atributos de calidad deseables que deben ser abordados con un buen diseño del software. La asignatura estará orientada a patrones. Se estudiará el concepto de arquitectura del software y estilos y patrones arquitectónicos. Se estudiará la asignación de responsabilidades en subsistemas y clases basada en patrones GRASP (*General Responsability ASsignment Patterns*). Por último se estudiarán el concepto de Patrones de Micro-arquitectura o Patrones de Diseño y se impartirá una colección de éstos extraídos de catálogos bien conocidos.

En la asignatura de especificarán los artefactos de entrada y salida de la fase de diseño. Estos serán principalmente modelos UML, código y pruebas. El alumno aprenderá a describir mediante diagramas UML tanto la arquitectura lógica del software, como su correspondencia con la arquitectura del sistema, mediante diagramas de despliegue. Especial énfasis se realizará en la descripción del comportamiento basada en diagramas de secuencia, tanto para la realización en diseño de los casos de uso como para explicar la interacción entre los objetos en la aplicación de un patrón de diseño. La parte práctica incluirá un apartado formativo en el uso de las herramientas y técnicas a emplear y, por otra parte, un trabajo práctico en equipo.

1.2 Relación con otras materias

La asignatura está planteada como una parte de las disciplinas que componen la Ingeniería de Software y que se desarrollan en las asignaturas de la materia del plan de estudios para la mención del mismo nombre en el módulo “Tecnologías Específicas”. Está situada en el segundo semestre del tercer curso. Por otro lado, la asignatura parte de los conocimientos impartidos en segundo curso en la asignatura Fundamentos de Ingeniería del Software, en la que se introduce al alumno el concepto de Proceso de Desarrollo y sus Fases, así como conceptos sobre desarrollo de calidad e incluso una introducción al modelado con UML y a los patrones GRASP.
Las asignaturas Fundamentos de Programación, Paradigmas de Programación y Programación Orientada a Objetos son relevantes de cara a que el alumno adquiera competencias en relación con la Programación Orientada a Objetos ya que la esencia de la asignatura es Diseño arquitectónico y diseño de alto nivel y detallado de software Orientado a Objetos.
La actividad de Diseño es una actividad integradora y muy técnica que incluye aspectos de conexión con bases de datos, diseño de interfaces gráficas, selección de estructuras de datos, etc., por lo que se beneficia de los conocimientos adquiridos por los alumnos en las asignaturas de segundo curso Interacción Persona-Computadora y Estructuras de Datos y Algoritmos y de la asignatura de primer semestre de tercer curso, Diseño de Bases de Datos.
Por último cabe indicar que esta asignatura puede considerarse continuadora directa de la asignatura de primer semestre de tercer curso “Modelado de Sistemas Software” y antecesora de la asignatura de primer cuatrimestre de cuarto curso “Planificación y Gestión de Proyectos”. La asignatura también puede considerarse una base para la asignatura “Desarrollo basado en Componentes y Servicios”. Se complementa adecuadamente con la asignatura optativa de primer cuatrimestre de tercer curso “Tecnologías para el desarrollo de Software”.

1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno haya cursado, e idealmente aprobado, las siguientes asignaturas:

- Programación Orientada a Objetos (primer semestre, segundo curso)
- Fundamentos de Ingeniería del Software (segundo semestre, segundo curso)
- Modelado de Sistemas Software (primer semestre, tercer curso)
- Diseño de Bases de Datos (primer semestre, tercer curso)

Se requiere una buena disposición para el trabajo en equipo.
Es recomendable disponer de un nivel de inglés que permita al estudiante leer bibliografía de consulta.

2. Competencias

2.1 Generales

Código	Descripción
	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática.
G03	Capacidad de análisis y síntesis
G04	Capacidad de organizar y planificar
G05	Comunicación oral y escrita en la lengua propia
G06	Conocimiento de una segunda lengua (preferentemente inglés)
G08	Habilidades de gestión de la información
G09	Resolución de problemas
G10	Toma de decisiones
G11	Capacidad crítica y autocrítica
G12	Trabajo en equipo
G14	Responsabilidad y compromiso ético
G15	Liderazgo
G16	Capacidad de aplicar los conocimientos en la práctica
G17	Habilidades de investigación
G18	Capacidad de aprender
G19	Capacidad de adaptarse a nuevas situaciones
G20	Capacidad de generar nuevas ideas
G21	Habilidad para trabajar de forma autónoma
G22	Diseño y gestión de proyectos

2.2 Específicas

Código	Descripción
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 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 las teorías, modelos y técnicas actuales.
CI3	Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.
CI7	Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más a la resolución de un problema.

CI8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CI13	Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web.
CI16	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la Ingeniería de Software.
CI17	Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

3. Objetivos

Código	Descripción
IS2.1	Conocer los principios y objetivos del software.
IS2.2	Evaluar diferentes alternativas de diseño en base a los principios y conceptos del diseño software.
IS2.3	Aplicar patrones de diseño adecuados para la construcción de una aplicación software.
IS2.4	Conocer las arquitecturas del software y su papel dentro de los procesos de diseño software.
IS2.5	Elaborar y documentar el diseño software para un producto software de tamaño reducido.
IS2.7	Aplicar técnicas específicas de modelado para sistemas de información.

4. Contenidos

1 Conceptos fundamentales
- 1.1 El diseño en el proceso de desarrollo.
- 1.2 Artefactos de entrada.
- 1.3 Artefactos de salida.
- 1.4 Diseño centrado en la arquitectura y centrado en casos de uso.

2 Calidad en el Diseño
- 2.1 Diseño Orientado a Objetos
- 2.2 Atributos externos de calidad: fiabilidad, mantenibilidad, usabilidad, desempeño.
- 2.3 Atributos internos de calidad: bajo acoplamiento, alta cohesión, ocultación de información, eficiencia.
- 2.4 Principios de Diseño

3 Arquitectura del Software
- 3.1 Patrones arquitectónicos: definición, catálogos
- 3.2 Cliente-Servidor, Cliente rico y Cliente delgado.
- 3.3 Pizarra, Repositorio, Filtros y Tuberías.
- 3.4 Capas,
- 3.5 MVC pasivo y activo,
- 3.6 Ejemplos de la arquitectura de referencia de algunos fabricantes.

4 Realización en Diseño de Casos de Uso
- 4.1 Repaso de patrones de asignación de responsabilidades
  - 4.1.1Experto, Creador, Controlador, Alta Cohesión, Bajo Acoplamiento.
  - 4.1.2 Fabricación pura, Variaciones protegidas, Indirección y Polimorfismo
- 4.2 Aplicación de los patrones GRASP en la realización en diseño de un caso de uso.
- 4.3 Modelo dinámico. Casos de estudio.

5 Patrones de micro-arquitectura
- 5.1 Introducción a los patrones de diseño
- 5.2 Patrones de diseño que permiten resolver patrones GRASP: adaptador, fachada, comando, patrones estado y estrategia.
- 5.3 Patrones de diseño: compuesto, iterador, singleton, método factoría y factoría abstracta
- 5.4 Patrones en interfaces de usuario: observador, decorador
- 5.5 Patrones de acceso a datos: Gateways, ActiveRecord, DAO+DTO, Broker, aplicación del patrón Proxy

6 Diseño detallado
- 6.1 Realización en diseño de relaciones
- 6.2 Selección de estructuras de datos
- 6.3 Documentación.

7. Despliegue
- 7.1 Correspondencia de la arquitectura lógica y la arquitectura del sistema
- 7.2 Diagramas de despliegue
- 7.3 Configuraciones

5. Métodos docentes y principios metodológicos

Actividad	Metodología
Clase de teoría	• Clase magistral participativa • Estudio de casos en aula • Resolución de problemas (realización en diseño de casos de uso, aplicación de patrones GRASP, arquitectónicos y de diseño o micro-arquitectura, etc.)
Clase práctica	• Se utilizará un método basado en la realización de un proyecto, siguiendo un esquema paralelo al de los casos de estudio presentados en el aula y siempre guiado por el profesor, que encargará y controlará el trabajo no presencial que se realizará en grupos (de 3 o 4 alumnos), siguiendo un enfoque colaborativo.
Seminarios Tutorías	• Talleres de aprendizaje del manejo de herramientas en sesiones específicas. • Seguimiento de las prácticas desarrolladas en grupo.

6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	50
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	40
Laboratorios (L)	28		
Prácticas externas, clínicas o de campo			
Seminarios (S)	2		
Tutorías grupales (TG)			
Evaluación			
Total presencial	**60**	**Total no presencial**	**90**

7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Entrega práctica	50	Coincidiendo con el examen de la convocatoria ordinaria con posibilidad de recuperar en convocatoria extraordinaria.
Examen sobre un supuesto práctico, resolución de ejercicios del tipo de los realizados en aula y laboratorio	35	Periodo de exámenes (ordinario y extraordinario).
Tests	15	A lo largo del cuatrimestre se realizarán 3 tests sobre los contenidos impartidos, cada uno con un peso del 5%, para un total del 15%.

Criterios de calificación

- **Calificación final**: Suma ponderada de las prácticas en parejas (50%), tests parciales (15%, 5% cada uno) y examen (35%), debiendo obtener una suma igual o mayor a 5. Será necesaria una calificación mínima de 4/10 en el examen para poder compensar con el resto de la evaluación.
    - Si nota(examen) >= 4, Nota final= Suma ponderada
    - Si nota(examen) < 4, Nota final= mínimo(Suma ponderada; 4,5)
- **Calificación de la convocatoria extraordinaria**: se utilizará la misma fórmula de cálculo de la nota final,
    - Obligatoriamente, se realizará un examen con el mismo formato que en la convocatoria ordinaria
    - Opcionalmente, los alumnos podrán volver a presentar la práctica que sustituiría a la entregada en convocatoria ordinaria.
    - Como caso excepcional, en aquellos equipos de trabajo que no han funcionado adecuadamente el alumno podrá optar a realizar el examen de convocatoria extraordinaria considerando la nota obtenida en éste como la nota de la asignatura.
    - Opcionalmente, los alumnos podrán elegir recuperar la nota obtenida en los tests haciendo un único test que abarque todos los temas evaluados en los tres tests realizados en convocatoria ordinaria.
- **Seminario voluntario:** Los alumnos que se ofrezcan como voluntarios para preparar e impartir un seminario sobre el diseño de un determinado framework con énfasis en la arquitectura y patrones que obliga o promueve dicho framework podrán ser reconocidos con 0,5 adicional en la nota final.

8. Consideraciones finales