



Guía docente de la asignatura

| | | | |
|--|---|----------------------|-------|
| Asignatura | DISEÑO DE SOFTWARE | | |
| Materia | DESARROLLO DE SOFTWARE | | |
| Módulo | | | |
| Titulación | Grado en INGENIERÍA INFORMÁTICA | | |
| Plan | 463 | Código | 45203 |
| Periodo de impartición | S5 | 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 → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingeniería Informática → Tutorías | | |
| 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 "Desarrollo de Software" (ver apartado 1.2), proporcionando en conjunto un enfoque de ingeniería de software a la titulación. La asignatura repasará el concepto de Proceso de Desarrollo para ubicar la etapa de Diseño en el proceso. Por otra parte, se repasarán 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 patrones arquitectónicos. Se estudiará la asignación de responsabilidades en subsistemas y clases basada en patrones GRASP (*General Responsibility 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 se especificarán los artefactos de entrada y salida de la fase de diseño. Estos serán principalmente modelos UML. 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. 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 “Desarrollo de Software”. Está situada en el segundo semestre, junto con la asignatura “Desarrollo basado en componentes y servicios”, pero en principio no se prevé una acción conjunta entre ambas asignaturas. Por otro lado, la asignatura parte de los conocimientos impartidos en Primer Curso en las asignaturas 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.

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”.

1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno haya aprobado la asignatura de Fundamentos de Ingeniería del Software, así como Fundamentos de Programación, ambas de primer curso. Sería deseable que el alumno haya cursado y superado las asignaturas del semestre anterior (primer semestre de tercer curso) Programación Orientada a Objetos, Modelado del Software y Diseño de Bases de Datos. 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 |
|--------|---|
| 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. 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 |





5. Bloques temáticos

Bloque 1: Modelado de sistemas software

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Después de una introducción general se estudian en profundidad el modelado de negocios y de requisitos, así como los modelos estáticos y dinámicos de sistemas software.

b. Objetivos de aprendizaje

| 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. |

c. 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
 - 4.1.1 Experto, 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: patrón 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: 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

d. Métodos docentes

| 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 (2/3 alumnos), siguiendo un enfoque colaborativo. |
| Seminarios Tutorías | Talleres de aprendizaje del manejo de herramientas en sesiones específicas. Seguimiento de las prácticas desarrolladas en grupo. |

e. Plan de trabajo

Ver cronograma al final

f. Evaluación

| INSTRUMENTO/PROCEDIMIENTO | PESO EN LA NOTA FINAL | OBSERVACIONES |
|--|-----------------------|--|
| Entrega práctica | 40 | 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 | 60 (global) | Periodo de exámenes (ordinario y extraordinario). |

g. Bibliografía básica

[Larman] Larman, C. "UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado". Prentice Hall, 2002. (2ª ed.)

[Arlow] Arlow, Jim, Neustadt, Ila. "UML 2", Anaya Multimedia, 2006.



[Gamma] Gamma, Helm, Johnson, Vlissides. "Patrones de Diseño", Addison Wesley

[Buschmann] Buschmann, Meunier, Rohnert, Sommerland, Stal, "Pattern-Oriented Software Architecture: A system of patterns". Wiley.

h. Bibliografía complementaria

[Booch-a] Booch, G., Jacobson, I., Rumbaugh, J. "El Lenguaje Unificado de Modelado. Guía del usuario". Addison-Wesley/Diaz de Santos, 2º edición, 2005

[Rumbaugh] Rumbaugh, J., Jacobson, I., Booch, G. "El Lenguaje Unificado de Modelado. Manual de referencia" 2º edición. Pearson, 2007

[Fowler] Fowler, M., "Patterns of Enterprise Application Architecture", Addison-Wesley Professional; 1era ed, 2002

[Bass] Len Bass, Paul Clements, Rick Kazman, "Software Architecture in Practice", Addison-Wesley Professional; 2da ed, 2003

[Gorton] Gorton, I. "Essential Software Architecture", Springer, 2ª ed., 2011

[Clements] Clements, P. et al, "Documenting Software Architectures: Views and Beyond", Springer, 2da ed., 2010

i. Recursos necesarios

Herramientas de ingeniería de software y estándares ISO/UML, instalados en los laboratorios del Centro y proporcionados a través de aula virtual.

6. Temporalización (por bloques temáticos)

| BLOQUE TEMÁTICO | CARGA ECTS | PERIODO PREVISTO DE DESARROLLO |
|-----------------|------------|--------------------------------|
| Bloque 1 | 6 | Semanas 1..16 |
| | | |
| | | |

Ver detalle en el cronograma de actividades

7. Tabla resumen de los instrumentos, procedimientos y sistemas de evaluación/calificación



| INSTRUMENTO/PROCEDIMIENTO | PESO EN LA NOTA FINAL | OBSERVACIONES |
|--|-----------------------|--|
| Entrega práctica | 40 | 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 | 60 (global) | Periodo de exámenes (ordinario y extraordinario). |

| Crterios de calificación |
|---|
| <p>Calificación final: Suma ponderada de las prácticas en parejas (40%) y examen (60%), debiendo obtener una suma igual o mayor a 5. Será necesaria una calificación mínima de 4/10 en el examen para poder compensar con prácticas.</p> <ul style="list-style-type: none"> o Si nota(examen) >= 4, Nota final= Suma ponderada o Si nota(examen) < 4, Nota final= mínimo(Suma ponderada; 4,5) <p>Calificación de la convocatoria extraordinaria: se utilizará la misma fórmula de cálculo de la nota final,</p> <ul style="list-style-type: none"> o Obligatoriamente, se realizará un examen con el mismo formato que en la convocatoria ordinaria o Opcionalmente, el alumno podrá volver a presentar la práctica que sustituiría a la entregada en convocatoria ordinaria. |

8. Consideraciones finales

9. Cronograma de actividades previstas

| Semana | Teoría | Prácticas |
|--------|------------|---|
| 1 | Tema 1 | Herramientas y método de trabajo: round-trip engineering, forward engineering, reverse-engineering |
| 2 | Tema 2 | La base de datos derby. El api jdbc, java.sql. Acceso a datos en local y en remoto. |
| 3 | Tema 3 | El diseñador de interfaz gráfica. El framework javax.swing. |
| 4 | 3 | Un ejemplo sencillo de arquitectura de 3 capas |
| 5 | 3 y Tema 4 | Añadir MVC en 3 capas. |
| 6 | 4 | Arquitectura de referencia para el trabajo de laboratorio: 3 capas + Cliente-Servidor con Cliente rico + MVC |
| 7 | 4 | Realización en diseño del Caso de Uso Identificarse En esta semana se presentará el trabajo práctico a realizar por equipos de 3 estudiantes |



| | | |
|----|--------|--|
| 8 | Tema 5 | Trabajo practico grupal con supervisión del profesor |
| 9 | 5 | Trabajo practico grupal con supervisión del profesor |
| 10 | 5 | Trabajo practico grupal con supervisión del profesor |
| 11 | 5 | Trabajo practico grupal con supervisión del profesor |
| 12 | 5 | Trabajo practico grupal con supervisión del profesor |
| 13 | 5 | Trabajo practico grupal con supervisión del profesor |
| 14 | 5 | Trabajo practico grupal con supervisión del profesor |
| 15 | Tema 6 | Trabajo practico grupal con supervisión del profesor |
| 16 | | Despliegue (Seminario) |

Nota: Sólo de carácter orientativo.

