



DESCRIPCIÓN DE LA ASIGNATURA

| | |
|---|--|
| Grado/Máster en: | Graduado/a en Ingeniería del Software por la Universidad de Málaga |
| Centro: | Escuela Técnica Superior de Ingeniería Informática |
| Asignatura: | Modelado y Diseño del Software |
| Código: | 303 |
| Tipo: | Obligatoria |
| Materia: | Proceso de Desarrollo de Software |
| Módulo: | Ingeniería del Software I |
| Experimentalidad: | 69 % teórica y 31 % práctica |
| Idioma en el que se imparte: | Español |
| Curso: | 3 |
| Semestre: | 1 |
| Nº Créditos | 6 |
| Nº Horas de dedicación del estudiante: | 150 |
| Nº Horas presenciales: | 60 |
| Tamaño del Grupo Grande: | 72 |
| Tamaño del Grupo Reducido: | 30 |
| Página web de la asignatura: | |

EQUIPO DOCENTE

Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

Área: LENGUAJES Y SISTEMAS INFORMÁTICOS

| Nombre y Apellidos | Mail | Teléfono Laboral | Despacho | Horario Tutorías |
|--|-----------|------------------|-------------------------------|---|
| Coordinador/a: ANTONIO JESUS VALLECILLO MORENO | av@uma.es | 952134297 | 3.2.48 - E.T.S.I. INFORMÁTICA | Todo el curso: Lunes 08:30 - 10:30, Jueves 10:30 - 12:30, Miércoles 10:30 - 12:30 |

RECOMENDACIONES Y ORIENTACIONES

Se recomienda cursar conjuntamente con la asignatura de Ingeniería de Requisitos, del mismo curso y semestre, o haber superado dicha asignatura con anterioridad. Asimismo, se recomienda haber superado previamente la asignatura de Introducción a la Ingeniería del Software, de segundo curso.

CONTEXTO

El objetivo de esta asignatura es presentar los principales conceptos, actividades y herramientas relacionados con el modelado y diseño de sistemas software y desarrollar la capacidad del alumno para llevar a cabo estas fases del proceso de desarrollo, así como las habilidades necesarias en el uso de herramientas a este efecto. Se seguirá un enfoque orientado a objetos, y basado en el estándar UML.

Las clases se desarrollarán tanto en el aula como en el laboratorio de ordenadores. Durante las mismas se irán presentando los distintos contenidos teóricos de la asignatura a la vez que se ejercitará su aplicación práctica y se mostrará el manejo de las herramientas informáticas relacionadas con ellos, avanzando de esta manera conjuntamente tanto en la adquisición y comprensión de contenidos teóricos como en el desarrollo de habilidades prácticas. La asistencia a las clases será obligatoria.

En particular, tras superar la asignatura el alumno tiene que:

- Conocer los conceptos y mecanismos básicos del modelado y diseño de software.
- Utilizar las notaciones existentes para el modelado y diseño de software.
- Aplicar métodos y herramientas de modelado orientado a objetos para la especificación y el diseño de sistemas de software.
- Realizar el modelado y diseño de un sistema software a partir de sus requisitos.

COMPETENCIAS

1 Competencias generales y básicas.

BÁSICAS

- CB02** 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.
- CB04** Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- CB05** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

GENERALES

- CG03** Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.



1 Competencias generales y básicas.

GENERALES

- CG05** 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, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias básicas, comunes y específicas del título.
- CG06** Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos según lo establecido en las competencias básicas, comunes y específicas del título.
- CG09** Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática
- CG11** Capacidad para analizar y valorar el impacto social y medioambiental de las soluciones técnicas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico en Informática.

2 Competencias específicas.

Competencias de Tecnología Especifica

- CE-IS-01** Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos 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.
- CE-IS-03** Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.
- CE-IS-04** 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.

CONTENIDOS DE LA ASIGNATURA

Nombre Bloque Temático

Unidad 1. Introducción al modelado conceptual.

Fundamentos y conceptos básicos. Descomposición, abstracción y generalización. Vistas. Sintaxis y semántica. Modelos matemáticos.

Unidad 2. Modelado estructural.

Modelado estático del sistema. Entidades y relaciones. Roles y cardinalidad. Restricciones. Tipos particulares de relaciones. Reificación y clases asociación. Especialización y generalización. Genericidad. Representación en UML. Diagramas de clases y de instancias. Diagramas de paquetes. Metamodelado y estereotipado.

Unidad 3. Modelado dinámico.

Modelado del funcionamiento del sistema. Relaciones entre los elementos durante la ejecución del sistema. Eventos del sistema y tipos de eventos. Acciones y efectos. Precondiciones y postcondiciones Consistencia con el modelo estructural. Representación en UML. Diagramas de interacción: de secuencia y de interacción. Diagramas de actividad y diagramas de estados.

Unidad 4. Modelado arquitectónico.

Modelado de la estructura del sistema. Estilos y patrones arquitectónicos. Subsistemas. Componentes. Interfaces y puertos. Vista de caja blanca y vista de caja negra. Nodos. Representación en UML. Diagramas de componentes. Diagramas de despliegue.

Unidad 5. Diseño orientado a objetos.

El diseño dentro del proceso de desarrollo: relación con las demás fases. Objetivos y resultados. Bases del diseño orientado a objetos. Revisión de las características del modelo de objetos más relevantes desde el punto de vista del diseño. Diseño e implementación de los modelos estático, dinámico y arquitectónico de los sistemas software.

Unidad 6. Patrones de diseño

Patrones de arquitectura y patrones de diseño.

Tipos de patrones de diseño. Patrones de creación, estructurales de comportamiento.

Unidad 7. Refactorización.

Principios de la refactorización de software. Refactorización y patrones de diseño. Catálogo de refactorizaciones.

ACTIVIDADES FORMATIVAS

Actividades presenciales

Actividades expositivas



Actividades presenciales

Actividades expositivas

Lección magistral

Actividades prácticas en instalaciones específicas

Prácticas en laboratorio

ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación no presenciales

Actividades de evaluación de la asignatura con participación alumnos

Cuestionario/encuesta

Actividades de evaluación presenciales

Actividades de evaluación del estudiante

Examen final

Realización de trabajos y/o proyectos

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

Los resultados del aprendizaje serán principalmente los siguientes:

RA1. Realizar el análisis de un sistema de software en base a los requisitos capturados

RA2. Conocer los conceptos y mecanismos básicos del modelado de sistemas software.

RA3. Manejar las notaciones existentes para el modelado y diseño de sistemas

RA4. Aplicar métodos y herramientas de modelado orientado a objetos para la especificación y el diseño de sistemas de software.

Estos resultados de aprendizaje se relacionan con las competencias específicas de la siguiente forma: la competencia CE-IS-01 se concreta en los resultados de aprendizaje RA1 y RA4, mientras que la competencia CE-IS-04 se concreta en los resultados de aprendizaje RA2 a RA4.

Se evaluará la correcta adquisición de dichos resultados de aprendizaje mediante la entrega de prácticas, acompañadas de su correspondiente memoria justificativa y los exámenes que se realizarán en las convocatorias oficiales. Estas actividades de evaluación permitirán también medir la adquisición de las competencias generales CG03, CG05, CG06 y CG11, relacionadas con la capacidad de concebir, diseñar y desarrollar sistemas software y estimar el impacto de las soluciones técnicas propuestas y las de resolver problemas con iniciativa y creatividad y tomar decisiones (CG09) y aplicar los conocimientos adquiridos y defender las soluciones propuestas, transmitir información y desarrollar habilidades de aprendizaje, recogidas en las competencias básicas CB2, CB4 y CB5.

PROCEDIMIENTO DE EVALUACIÓN

Durante las clases se plantearán a los alumnos una serie de prácticas (entre cuatro y cinco), relacionadas con los contenidos de la asignatura, por medio de las cuales los alumnos pueden tanto adquirir y fijar las aptitudes en el manejo de los conceptos teóricos y herramientas informáticas presentadas, como demostrar las habilidades adquiridas. Cada una de estas prácticas será planteada y comenzada en el laboratorio, pero completada por los alumnos mediante su trabajo personal. Las prácticas se harán en grupos reducidos, cuyo tamaño estará en función del número de alumnos matriculados. Las prácticas tendrán carácter obligatorio y serán evaluadas y calificadas por el profesor, quien fijará las fechas de entrega de las mismas. Dichas fechas serán anteriores a la fecha fijada en la planificación docente de la ETSI Informática para el examen de la primera convocatoria ordinaria de la asignatura. En caso necesario, la individualización de la nota de prácticas se realizará en base a pruebas individuales, encuestas y entrevistas con los alumnos. Como resultado de todo ello, cada alumno obtendrá una nota de prácticas (NPR), en escala 0..10.

De manera regular (aproximadamente cada dos semanas), se plantearán también a los alumnos diversas cuestiones y ejercicios, de carácter más breve que las prácticas ya mencionadas. El objetivo es que los alumnos discutan y resuelvan dichas cuestiones y ejercicios de manera colaborativa, a través del campus virtual, para que de esta manera reflexionen y puedan fijar mejor los conceptos teóricos y habilidades prácticas presentados a lo largo del curso. La asistencia y participación en las clases y en la resolución de los cuestionarios y ejercicios planteados a través del campus virtual, se evaluará también con una nota (NQE), en escala 0..1.

Al final del semestre se realizará un examen escrito, incluyendo tanto aspectos teóricos como prácticos. El alumno deberá demostrar en dicha prueba que ha asimilado los conceptos impartidos a lo largo del curso, así como que ha desarrollado las correspondientes habilidades prácticas en el modelado y diseño de sistemas software. Como resultado, cada alumno obtendrá una calificación de examen (NEX), en escala 0..10.

Los alumnos que alcancen una calificación NPR + NQE de 7 puntos, podrán optar a una versión reducida del examen escrito. Por el contrario, los alumnos que no alcancen la citada calificación deberán realizar el examen completo.

Atendiendo a lo anterior, la calificación final del alumno se calculará en base a la siguiente fórmula:

$$\text{Calificación} = \text{NEX} * (1 + (\text{NPR} + \text{NQE} - 5) / 10)$$



Dicha calificación será truncada a 10 en caso necesario. Para poder superar la asignatura, el alumno deberá obtener una calificación mínima NEX de 3.5 sobre 10 puntos en el examen escrito.

Finalmente, los alumnos que no alcancen los 5 puntos sumando sus notas de prácticas y de cuestiones y ejercicios ($NPR + NQE < 5$), deberán realizar la versión completa del examen escrito, siendo en este caso el umbral necesario para superar la asignatura el habitual de 5 puntos sobre 10 en dicho examen.

En la convocatoria de septiembre y en las convocatorias extraordinarias, los alumnos deberán realizar la versión completa del examen escrito, siendo su calificación final la que obtengan en dicho examen. En este caso el umbral necesario para superar la asignatura es el habitual de 5 puntos sobre 10.

Aquellos alumnos que tengan reconocida la condición de estudiante a tiempo parcial o deportista universitario de alto nivel tendrán prioridad a la hora de elegir horario de grupo reducido y se tendrá en cuenta su situación a la hora de flexibilizar los criterios referidos a asistencia a las clases y en particular a las prácticas y/o exámenes.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

- Applied Java Patterns; S. Stelting, O. Maassen; Prentice Hall; 2002
- Applying UML and Patterns: an introduction to Object-Oriented Analysis and Design and the Unified Process; C. Larman; Prentice-Hall; 2002
- Conceptual Modeling of Information Systems; A. Olivé; Springer; 2007
- Design Patterns; E. Gamma et al.; Addison Wesley; 1995
- El Lenguaje Unificado de Modelado: Guía del Usuario; G. Booch, J. Rumbaugh, I. Jacobson; Addison Wesley; 2000
- El Lenguaje Unificado de Modelado: Manual de Referencia; G. Booch, J. Rumbaugh, I. Jacobson; Addison Wesley; 2000
- El Proceso Unificado de Desarrollo de Software; G. Booch, J. Rumbaugh, I. Jacobson; Addison Wesley; 2000
- Object-Oriented Systems Analysis and Design Using UML; S. Bennett et al.; McGraw Hill; 2002
- Refactoring: improving the design of existing code; M. Fowler; Addison Wesley; 2000
- The Object Primer. Agile Model-Driven development with UML 2.0, (3rd edition).; S.W. Ambler; Cambridge University Press; 2009

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

| Descripción | Horas | Grupo grande | Grupos reducidos |
|--|------------|-------------------------------------|-------------------------------------|
| Lección magistral | 41,4 | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| Prácticas en laboratorio | 18,6 | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL | 60 | | |
| TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL | 75 | | |
| TOTAL HORAS ACTIVIDAD EVALUACIÓN | 15 | | |
| TOTAL HORAS DE TRABAJO DEL ESTUDIANTE | 150 | | |

