

Asignatura: Metodologías, Desarrollo y Calidad en la Ingeniería de Software

Cuatrimestre: 1º

ECTS: 6

Carácter: OB

Contenidos:

ID	Descripción
C1	Ingeniería de Software avanzada, OO, componentes y Web.
C2	Diseño avanzado de software.
C3	Desarrollo de software: nuevos paradigmas de programación, patrones y frameworks.
C4	Metodologías avanzadas en Ingeniería de Software: agilidad.
C5	Procesos de calidad del software, métricas, modelos de madurez y mejora del software.

Competencias¹:

- CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.
- CB7- Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.
- CB8 - Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.
- CB9 - Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades.
- CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

¹ CB: Competencia básica; CG: Competencia general; CE: Competencia específica ; CT: Competencia transversal

- CG1 - Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en el ámbito de la Ingeniería de Software.
- CG2 - Capacidad para dirigir, planificar y supervisar equipos multidisciplinares.
- CG3 - Capacidad para la elaboración, planificación estratégica, dirección, coordinación y gestión técnica y económica de proyectos en todos los ámbitos de la Ingeniería de Software siguiendo criterios de calidad.
- CG4 - Capacidad para la dirección general, dirección técnica y dirección de proyectos de I+D+I, en empresas y centros tecnológicos, en el ámbito de la Ingeniería de Software.
- CG5 - Capacidad para la puesta en marcha, dirección y gestión de procesos de diseño y desarrollo de sistemas informáticos, con garantía de la seguridad para las personas y bienes, la calidad final de los productos y su homologación.
- CG6 - Capacidad para aplicar los principios de la economía y de la gestión de recursos humanos y proyectos, así como la legislación, regulación y normalización de la informática.
- CE1 - Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, sistemas, servicios y contenidos informáticos.
- CE2 - Capacidad para utilizar y desarrollar metodologías, métodos, técnicas, programas de uso específico, normas y estándares de Ingeniería de Software.
- CE3 - Capacidad para analizar las necesidades de información que se plantean en un entorno y llevar a cabo en todas sus etapas el proceso de construcción de un sistema de información.
- CE4 - Capacidad para crear y diseñar sistemas software que resuelvan problemas del mundo real.
- CE5 - Capacidad para evaluar y utilizar entornos de Ingeniería de Software avanzados, métodos de diseño, plataformas de desarrollo y lenguajes de programación.
- CE7 - Capacidad para la dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, con garantía de la seguridad para las personas y bienes, la calidad final de los productos y su homologación.
- CT1 -Analizar de forma reflexiva y crítica las cuestiones más relevantes de la sociedad actual para una toma de decisiones coherente.
- CT2 -Identificar las nuevas tecnologías como herramientas didácticas para el intercambio comunicacional en el desarrollo de procesos de indagación y de aprendizaje grupal.
- CT3 - Aplicar los conocimientos y capacidades aportados por los estudios a casos reales y en un entorno de grupos de trabajo en empresas u organizaciones.
- CT4 - Adquirir la capacidad de trabajo independiente, impulsando la organización y favoreciendo el aprendizaje autónomo.

Metodologías docentes:

ID	Denominación	Descripción
MD1	Lección magistral	Presentación de un tema estructurado para facilitar los contenidos sobre la materia objeto de estudio de forma organizada.
MD2	Estudios de casos	Análisis de un problema o suceso real para conocerlo, interpretarlo, resolverlo, generar hipótesis, contrastar datos, reflexionar, completar conocimientos, diagnosticarlo y buscar las soluciones.
MD3	Resolución de ejercicios y problemas	Ejercitar, ensayar y poner en práctica los conocimientos previos. Suele utilizarse como complemento de la lección magistral.
MD4	Aprendizaje Basado en Problemas (ABP)	A partir de un problema diseñado por el profesor, el estudiante ha de resolverlo para desarrollar determinadas competencias previamente definidas.
MD5	Contrato de Aprendizaje	Acuerdo establecido entre el profesor y el estudiante para la consecución de unos aprendizajes a través de una propuesta de trabajo autónomo, con la supervisión del profesor.

Temario:

Tema 1. Introducción a la Ingeniería de Software

- 1.1. Introducción
- 1.2. La crisis del software
- 1.3. Diferencias entre la Ingeniería de Software y la Ciencia de la Computación
- 1.4. Ética y responsabilidad profesional en la ingeniería del software
- 1.5. Fábricas de software

Tema 2. El proceso de desarrollo de software

- 2.1. Definición
- 2.2. Modelo de proceso software
 - 2.2.1. Modelo en cascada
 - 2.2.2. Modelo en espiral
 - 2.2.3. Ingeniería de software basada en componentes
- 2.3. El proceso unificado de desarrollo de software

Tema 3. Desarrollo de software orientado a objetos

- 3.1. Introducción

- 3.2. Principios de la orientación a objetos
- 3.3. Definición de objeto
- 3.4. Definición de clase
- 3.5. Análisis orientado a objetos vs diseño orientado a objetos

Tema 4. Desarrollo de software basado en modelos

- 4.1. La necesidad de modelar
- 4.2. Modelado de sistemas software
- 4.3. Modelado de objetos
- 4.4. UML
- 4.5. Herramientas CASE

Tema 5. Modelado de aplicaciones con UML

- 5.1. Modelado avanzado de requisitos
- 5.2. Modelado estático avanzado
- 5.3. Modelado dinámico avanzado
- 5.4. Modelado de componentes

Tema 6. Patrones de diseño con UML

- 6.1. Introducción
- 6.2. Adapter
- 6.3. Factory
- 6.4. Singleton
- 6.5. Strategy
- 6.6. Composite
- 6.7. Facade
- 6.8. Observer

Tema 7. Ingeniería dirigida por modelos

- 7.1. Introducción
- 7.2. Metamodelado de sistemas
- 7.3. MDA
- 7.4. DSLs
- 7.5. Refinamiento de modelos con OCL

7.6. Transformaciones de modelos

7.6.1. Transformaciones modelo-a-modelo

7.6.2. Transformaciones modelo-a-texto

Tema 8. Ontologías en la Ingeniería de Software

8.1. Introducción

8.2. Ingeniería de la Ontología

8.3. Aplicación de las Ontologías en la Ingeniería de Software

Tema 9. Metodologías ágiles para el desarrollo de software

9.1. ¿Qué es agilidad en el software?

9.2. El manifiesto ágil

9.3. La hoja de ruta de un proyecto ágil

9.4. El “Product Owner”

9.5. Las historias de usuario

9.6. Planificación y estimación ágil

9.7. Métricas para desarrollos ágiles

Tema 10. Scrum

10.1. Principios de Scrum

10.2. Los roles

10.3. El product backlog

10.4. El sprint

Tema 11. Lean Software Development

11.1. Principios de “Lean Software Development”

11.2. Kanban

Tema 12. Calidad y mejora del proceso de software

12.1. Medición del software

12.2. Modelo de calidad de procesos software: CMMI

Bibliografía básica:

Aparte de los apuntes del profesor, se recomienda la siguiente bibliografía:

Booch, G., Rumbaugh, J. & Jacobson, I. (2006). *El Lenguaje Unificado de Modelado. UML 2.0 2ª Edición*. España: Pearson Addison-Wesley.

García Molina, J., García Rubio, F.O., Pelechano, V., Vallecillo, A., Vara, J.M. & Vicente-Chicote, C. (2013). *Desarrollo de Software Dirigido por Modelos: Conceptos, Métodos y Herramientas*. Ra-Ma.

Larman, C. (2003). *UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. Segunda edición*. Pearson Prentice-Hall.

Piattini, M., García, F.O., Garzás, J. & Fabiana Genero, M. (2008). *Medición y estimación del software*. Ra-Ma.

Piattini, M. & Garzás, J. (2010). *Fábricas de Software: Experiencias, Tecnologías y Organización. 2ª edición actualizada*. Ra-Ma.

Rumbaugh, J., Jacobson, I. & Booch, G. (2007). *El Lenguaje Unificado de Modelado. Manual de Referencia. UML 2.0 2ª Edición*. España: Pearson Addison-Wesley.

Sommerville, I. (2005). *Ingeniería del Software. Séptima edición*. España: Pearson Addison-Wesley.