Guía docente
270015 - IES - Introducción a la Ingeniería del Software

Última modificación: 30/01/2024

Unidad responsable: Facultad de Informática de Barcelona
Unidad que imparte: 747 - ESSI - Departamento de Ingenieria de Servicios y Sistemas de Información.

Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2010). (Asignatura obligatoria).

Curso: 2023 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: ERNEST TENIENTE LOPEZ

Otros:

Primer quadrimestre:
MARIA DOLORS COSTAL COSTA - 43
JOSE FRANCISCO CRESPO SANJUSTO - 12, 13
MONTSERRAT ESTAÑOL LAMARCA - 41, 42
MANUEL RELLO SALTOR - 11, 12, 13, 41, 42, 43
ERNEST TENIENTE LOPEZ - 11

Segon quadrimestre:
CLAUDIA PATRICIA AYALA MARTINEZ - 14, 41, 42, 43
JORDI CASANOVAS MUÑOZ - 22
MARIA DOLORS COSTAL COSTA - 31, 34
JOSE FRANCISCO CRESPO SANJUSTO - 11, 32
MONTSERRAT ESTAÑOL LAMARCA - 12, 13
SERGIO MORALES GARCIA - 41, 42
XAVIER ORIOL HILARI - 21, 22, 23, 43
MANUEL RELLO SALTOR - 11, 12, 13, 14, 31, 32, 33, 34
ERNEST TENIENTE LOPEZ - 21, 33

CAPACIDADES PREVIAS

Los alumnos deberían tener conocimientos suficientes de estructuras de datos y algoritmos y de programación cuando empiecen ingeniería del software.

REQUISITOS

- Pre-requisito BD
- Pre-Corequisito EDA

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
CT2.1. Demostrar conocimiento y capacidad de aplicación de los principios, metodologías y ciclos de vida de ingeniería de software.
CT2.3. Diseñar, desarrollar, seleccionar y evaluar aplicaciones, sistemas y servicios informáticos, y al mismo tiempo asegurar su fiabilidad, su seguridad y su calidad, conforme a principios éticos y a la legislación y la normativa vigente.
CT2.4. Demostrar conocimiento y capacidad de aplicación de las herramientas necesarias para el almacenaje, el procesamiento y el acceso a los Sistemas de información, incluidos los basados en web.
CT3.3. Saber encontrar e interpretar la información básica que permite evaluar el entorno económico de la organización.
CT5.5. Usar las herramientas de un entorno de desarrollo de software para crear y desarrollar aplicaciones.
CT6.1. Demostrar conocimiento y tener capacidad para administrar y mantener sistemas, servicios y aplicaciones informáticas.
CT7.1. Demostrar conocimiento de las métricas de calidad y saber utilizarlas.
CT8.1. Identificar tecnologías actuales y emergentes y evaluar si son aplicables, y en qué medida, para satisfacer las necesidades de los usuarios.
CT8.2. Asumir los roles y las funciones del jefe del proyecto y aplicar, en el ámbito de las organizaciones las técnicas de gestión y programación del tiempo, de los costes y de los aspectos financieros, de los recursos humanos y del riesgo.
CT8.3. Demostrar conocimiento y saber aplicar las técnicas apropiadas para modelar y analizar los diferentes tipos de decisiones.
CT8.4. Elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y la normativa vigente.
CT8.5. Gestionar y resolver los problemas y conflictos gracias a la capacidad de generar alternativas o escenarios de futuro convenientemente analizados, integrando los aspectos de incertidumbre y los múltiples objetivos a considerar.
CT8.6. Demostrar comprensión de la importancia de la negociación, de los hábitos de trabajo efectivos, del liderazgo y de las habilidades de comunicación en todos los entornos de desarrollo de software.
CT8.7. Controlar versiones y configuraciones del proyecto.

Genéricas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

METODOLOGÍAS DOCENTES

La docencia de la asignatura está estructurada en clases de teoría y clases de problemas.

En las clases de teoría los profesores presentan los contenidos esenciales de la asignatura. Normalmente los profesores usan transparencias, que los estudiantes pueden obtener, y deberían hacerlo, antes de las clases.

En las clases de problemas se practican los contenidos de la asignatura (los presentados en clase y los adquiridos autónomamente) mediante la realización de problemas. A veces se requiere que los problemas se hayan resuelto (o al menos intentado resolver) antes de la clase, y entonces la clase se destina a analizar y discutir colectivamente las mejores soluciones. Otras veces, el problema se plantea y se ha de resolver en la misma clase.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

1.Tener una visión general del proceso de la ingeniería del software
2.Ser capaces de entender los requisitos de un sistema software y relacionarlos con partes de las especificaciones
3.Ser capaces de escribir las especificaciones de un sistema software en UML
4.Conocer las propiedades deseables de las especificaciones de un sistema software
5.Ser capaces de analizar la completitud y la consistencia de las especificaciones
6.Conocer los principios generales de la arquitectura del software y del modelo de diseño orientado a objetos en UML
7.Comprender la estructura lógica en capas de un sistema de información: capa de presentación, de dominio y de gestión de datos
8.Saber transformar un modelo de especificación UML en uno de diseño
9.Conocer el concepto de patrón de diseño y dominar el uso de algunos de los patrones más conocidos
10.Conocer los conceptos básicos de la prueba del software

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas actividades dirigidas	6,0	4.00
Horas aprendizaje autónomo	84,0	56.00
Horas grupo mediano	30,0	20.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

Introducción a la ingeniería del software

Requisitos de un sistema software y su especificación

El Modelo de Casos de Uso en UML

Especificación de sistemas software con UML

Descripción:
- El esquema estructural de los datos
- El esquema del comportamiento
- Los diagramas de estado

Introducción al diseño de software

Diseño orientado a objetos con UML

ACTIVIDADES

Estudio de la introducción a la ingeniería del software

Objetivos específicos:
1

Dedicación: 4h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 2h

Estudio de los requisitos del software y de su relación con la especificación de sistemas software

Objetivos específicos:
1, 2

Dedicación: 3h
Grupo grande/Teoría: 1h
Aprendizaje autónomo: 2h

Estudio del Modelo de Casos de Uso en UML

Objetivos específicos:
2, 3

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 5h
Grupo grande/Teoría: 1h
Aprendizaje autónomo: 4h

Estudio del esquema estructural de los datos en UML

Objetivos específicos:
2, 3, 4, 5

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 34h
Grupo grande/Teoría: 4h
Grupo mediano/Prácticas: 10h
Aprendizaje autónomo: 20h

Estudio del modelo del comportamiento en UML

Objetivos específicos:
2, 3, 4, 5

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 28h
Grupo grande/Teoría: 4h
Grupo mediano/Prácticas: 8h
Aprendizaje autónomo: 16h

Estudio de la introducción al diseño de software

Objetivos específicos:
6, 7

Dedicación: 4h
Grupo grande/Teoría: 2h
Aprendizaje autónomo: 2h

Estudio del diseño de software

Objetivos específicos:
6, 8, 9

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 48h
Grupo grande/Teoría: 8h
Grupo mediano/Prácticas: 10h
Aprendizaje autónomo: 30h

Ejercicio del modelo estructural UML

Descripción:
Se evaluarán parte de los objetivos 3 a 5

Objetivos específicos:
3, 4, 5

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 5h
Actividades dirigidas: 2h
Aprendizaje autónomo: 3h

Control del modelo estructural en UML

Descripción:
Se evaluarán parte de los objetivos 1 a 5

Objetivos específicos:
1, 2, 3, 4, 5

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 5h
Actividades dirigidas: 2h
Aprendizaje autónomo: 3h

Control del modelo del comportamiento en UML

Descripción:
Se evaluarán parte de los objetivos 1 a 5

Objetivos específicos:
2, 3, 4, 5

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 5h
Actividades dirigidas: 2h
Aprendizaje autónomo: 3h

Ejercicio de diseño de software

Descripción:
Se evaluarán parte de los objetivos 8 y 9

Objetivos específicos:
8, 9

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 5h
Actividades dirigidas: 2h
Aprendizaje autónomo: 3h

Control del diseño de software

Descripción:
Se evaluarán parte de los objetivos 6 a 10

Objetivos específicos:
6, 7, 8, 9, 10

Competencias relacionadas:
G4. COMUNICACIÓN EFICAZ ORAL Y ESCRITA: Comunicar de forma oral y escrita con otras personas conocimientos, procedimientos, resultados e ideas. Participar en debates sobre temas propios de la actividad del ingeniero técnico en informática.

Dedicación: 4h
Actividades dirigidas: 2h
Aprendizaje autónomo: 2h

SISTEMA DE CALIFICACIÓN

Durante el curso se harán cinco actos evaluatorios: tres controles en horario de clase (C1, C2 i C3) y dos pruebas fuera del horario de clase (FHC1 y FHC2). Para el cálculo de la nota final también se tendrà en cuenta la participación activa en clase. La distribución secuencial de estos actos evaluatorios y la su contribución a la nota final de la assignatura vienen definidos por la fórmula siguiente:

Nota Final = 10% C1 + 30% FHC1 + 15% C2 + 10% C3 + 30% FHC2 + 5% partticipación

teniendo en cuenta que:
El control C3 es opcional. Si un alumno no realiza el control C3, entonces automáticamente el peso de la prueba FHC2 pasa a ser del 40%.
La nota de participación se consigue entregando un mínimo del 75% de los ejercicios propuestos y participando activamente en clase.
Nota Final = NP si el alumno no se presenta a la FHC2 y su nota final acumulada es

No hay examen final.

Además de la nota de la asignatura, habrá una nota a parte de la competencia genérica asignada a la asignatura, con valores A, B, C, D o NA donde:
o A indica que la competencia se ha superado con un nivel de excelencia
o B indica que la competencia se ha superado con el nivel deseado
o C indica que la competencia se ha superado con un nivel suficiente
o D indica que la competencia no se ha superado
o NA indica que la competencia no se ha evaluado

BIBLIOGRAFÍA

Básica:
- Larman, C. Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development. 3rd ed. Prentice Hall PTR, 2005. ISBN 0131489062.
- Pressman, R.S.; Maxim, B.R. Software engineering: a practitioner's approach. 9th ed. New York: McGraw Hill Higher Education, 2020. ISBN 9781260548006.
- Rumbaugh, J.; Jacobson, I.; Booch, G. The unified modeling language reference manual. 2nd ed. Addison-Wesley, 2005. ISBN 0321245628.
- Professorat d'IES. Transparències d'Introducció a l'Enginyeria del Software.
- Professorat d'IES. Exercicis i exercicis resolts d'Introducció a l'Enginyeria del Software.

Complementaria:
- Olivé, A. Conceptual modeling of information systems. Springer, 2007. ISBN 9783540393894.
- Olivé, A. Conceptual modeling of information systems. Berlin: Springer, 2007. ISBN 9783540393894.
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. Design patterns: elements of reusable object-oriented software. Addison-Wesley, 1995. ISBN 0201633612.
- Binder, R.V. Testing object-oriented systems: models, patterns, and tools. Addison-Wesley, 2000. ISBN 9780321700674.

RECURSOS

Enlace web:
- http://www.uml.org/- http://www.omg.org/- http://hillside.net/patterns/