

CENTRO: 180 - Escuela de Ingeniería Informática

TITULACIÓN: 4008 - Grado en Ingeniería Informática

ASIGNATURA: 40822 - INGENIERÍA DEL SOFTWARE II

Vinculado a : (Titulación - Asignatura - Especialidad)

4801-Doble Grado en Ingeniería Informática y - 48126-INGENIERIA DE SOFTWARE II - 00

CÓDIGO UNESCO: 40822

TIPO: Obligatoria

CURSO: 3

SEMESTRE: 1º semestre

CRÉDITOS ECTS: 6

Especificar créditos de cada lengua:

ESPAÑOL: 6

INGLÉS: 0

SUMMARY

REQUISITOS PREVIOS

Es requisito fundamental que los estudiantes hayan cursado las asignaturas de

1. Fundamentos de Programación, Programación I y Programación II
2. Ingeniería del Software I
3. Bases de datos I.

Datos identificativos del profesorado que la imparte.

Plan de Enseñanza (Plan de trabajo del profesorado)

Contribución de la asignatura al perfil profesional:

El estudiante que curse en este curso puede desarrollar las siguientes habilidades y capacidades:

Capacidad para hacer diseño y programación orientada a objetos

Capacidad de un alto grado de reutilización del código

Capacidad para la construcción de librerías

Habilidades para utilizar herramientas CASE (UML)

Capacidad para aplicar los fundamentos y los principios del diseño, seleccionar patrones de diseño para resolver un problema concreto.

Capacidad para plantear arquitecturas flexibles orientadas al cambio constante.

Competencias que tiene asignadas:

G1. Poseer y comprender conocimientos en un área de estudio (Ingeniería Informática) que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio

G2. 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

G3. Reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

N1. Comunicarse de forma adecuada y respetuosa con diferentes audiencias (clientes, colaboradores, promotores, agentes sociales, etc.), utilizando los soportes y vías de comunicación más apropiados (especialmente las nuevas tecnologías de la información y la comunicación) de modo que pueda llegar a comprender los intereses, necesidades y preocupaciones de las personas y organizaciones, así como expresar claramente el sentido de la misión que tiene encomendada y la forma en que puede contribuir, con sus competencias y conocimientos profesionales, a la satisfacción de

esos intereses, necesidades y preocupaciones.

N2. Cooperar con otras personas y organizaciones en la realización eficaz de funciones y tareas propias de su perfil profesional, desarrollando una actitud reflexiva sobre sus propias competencias y conocimientos profesionales y una actitud comprensiva y empática hacia las competencias y conocimientos de otros profesionales

N3. Contribuir a la mejora continua de su profesión así como de las organizaciones en las que desarrolla sus prácticas a través de la participación activa en procesos de investigación, desarrollo e innovación.

N4. Comprometerse activamente en el desarrollo de prácticas profesionales respetuosas con los derechos humanos así como con las normas éticas propias de su ámbito profesional para generar confianza en los beneficiarios de su profesión y obtener la legitimidad y la autoridad que la sociedad le reconoce.

T1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

T2. Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática, de acuerdo con los conocimientos adquiridos según lo establecido en apartado 5 de la resolución indicada.

T3. 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.

T4. Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.

T5. 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.

T8. Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

T10. Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

T11. 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.

CII01. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

CII02. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.

CII16. Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería

de software.

CII17. Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

IS01 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.

IS03 Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

IS04 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.

IS05 Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse.

Objetivos:

Aprender a diseñar software orientado a objetos

Conseguir que el estudiante adquiera las capacidades para abordar un diseño de software y su posterior implementación

Aprender a describir el diseño del software usando métodos y técnicas de representación

Aprender a refactorizar el código cuando se producen cambios en los requisitos

Aprender a usar los estándares y patrones de diseño

Aprender a gestionar las versiones del código

Contenidos:

1. Estructura del software

Definición y evolución

Programación estructurada

Estructura modular

Arquitecturas de software

Estilos de software

Patrones de diseño

Bibliografía. [PRESMAN]

Competencias: G1 G2 G3 T8 T11 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

2. Descomposición modular

Modularidad

Tipos de módulos

Interfaz e implementación

Cohesión y dependencia

Jerarquía de control

Bibliografía. [PRESMAN] [CRAIG]

Competencias: G1 G2 G3 T8 T11 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

3. Dependencias

Malos diseños

Inyección de dependencias

Inversión de dependencias

Inversion del control

Bibliografía. [PRESMAN][CRAIG]

Competencias: G1 G2 G3 T8 T11 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

4. Responsabilidades

- Reglas de diseño

- Responsabilidad única

- Principio abierto-cerrado

Bibliografía. [PRESMAN][CRAIG]

Competencias: G1 G2 G3 T8 T11 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

5. Abstracción

- Formación de conceptos

- Generalización

- Sustituibilidad

- Principio de sustitución de Liskov

- Diseño por contrato

Bibliografía. [PRESMAN][CRAIG]

Competencias: G1 G2 G3 T8 T11 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

6. Patrones de diseño

- Patrones creacionales

- Patrones de estructura

- Patrones de comportamiento

Bibliografía. [GAMMA]

Competencias: G1 G2 G3 T8 T11 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

Práctico (presencial)

1. Gestión de versiones del código

Competencias: N2 N3 T2 T4 T5 CII02 CII16 IS01 IS04

2. Desarrollo del modelo

Competencias: N1 N2 N3 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04

3. Arquitectura Controlador, Vista, Modelo

Competencias: N1 N2 N3 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04

4. Principios y patrones de diseño

Competencias: N1 N2 N3 N4 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

5. Arquitectura Presenter

Competencias: N1 N2 N3 N4 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04

Trabajo (no presencial)

1. Diseño de una aplicación con la arquitectura MVC

Competencias: CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

2. Diseño de una aplicación software con la arquitectura MVP

Competencias: CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

Metodología:

La asignatura se desarrollará mediante la realización de diversas actividades, clases en el aula, tanto teóricas como prácticas, seminarios de problemas en pequeños grupos, tanto en el aula como en el laboratorio, tutorías individualizadas, pruebas parciales y examen final.

Todas las actividades tienen como objetivo principal potenciar el aprendizaje de los alumnos, facilitando la adquisición de conocimientos y competencias. Las diferentes actividades estarán sujetas a un proceso de evaluación continua, y algunas permitirán certificar el aprendizaje.

A continuación se detallan las diferentes actividades que se realizarán a lo largo del curso en el ámbito de la asignatura.

Clases

La teoría básica necesaria será expuesta en las clases teóricas por el profesor de la asignatura.

En las clases prácticas en aula se realizarán ejercicios con diferentes grados de complejidad. En las clases prácticas del laboratorio el alumno realizará ejercicios en los que tenga que participar activamente.

Trabajos

Se realizarán dos trabajos propuestos por el profesor que consistirán en la resolución de ejercicios prácticos relacionados con la materia estudiada. El informe de cada trabajo deberá ir firmado por su autor y será revisado y valorado por el profesor, tanto en contenidos como en presentación, pudiendo ser requeridas por parte de los alumnos cuantas explicaciones se consideren oportunas. Cada alumno tendrá acceso a su informe, debidamente revisado y valorado.

Seminarios

Se realizarán dos seminarios de ejercicios al final del curso, cada uno de una hora de duración. Estos seminarios se realizarán en el aula y consistirán en la realización de ejercicios presentados por el profesor. Cada alumno entregará por escrito los resultados obtenidos, pudiendo ser requeridas por parte del profesor cuantas explicaciones se consideren oportunas para su evaluación.

Tutorías

Las tutorías individualizadas podrán ser atendidas los días y horas que el profesor establezca al inicio del curso.

Será en horas de tutorías cuando los alumnos reciban los trabajos, tras su revisión y evaluación por el profesor. Se indicará el horario de cada actividad oportunamente.

En todas las actividades realizadas se llevará a cabo un control de asistencia.

Criterios y fuentes para la evaluación:

Examen escrito final

Se realizará un examen escrito final relacionado con la materia estudiada, tanto teórica como práctica.

Competencias: N1 N2 N3 N4 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

Ejercicios y trabajos

Los ejercicios entregados por los alumnos en los seminarios y los trabajos contribuyen a definir la nota de la evaluación continua.

Competencias: N1 N2 N3 N4 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

Asistencia a clase

La asistencia a clase contribuye a definir la nota de la evaluación continua. El control de asistencia consiste en que el alumno en cada clase debe responder por escrito preguntas relacionadas con la clase de ese día. Este control lo entrega firmado el alumno para su posterior contabilización.

Competencias: N1 N2 N3 N4 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

Participación

La participación en clase la registra el profesor firmando el control de asistencia del alumno.

Competencias: N1 N2 N3 N4 T1 T2 T3 T4 T5 T10 CII01 CII02 CII16 CII17 IS01 IS03 IS04 IS05

Sistemas de evaluación:

Se establecen dos sistemas de evaluación:

- Para la prueba ordinaria, una evaluación continua basada en la valoración de la actividad en los seminarios y de los trabajos realizados por el alumno, así como un examen al final del curso. En estos casos, la actividad en clase representa el 40% de la calificación final y el 60% restante la nota del examen.
- Para las pruebas extraordinarias, se realizará un único examen que representará el 100% de la nota final.

En el sistema de evaluación continua se valorará fundamentalmente la asistencia a las clases teóricas y prácticas, la entrega de los trabajos y la participación en la resolución de los ejercicios que el profesor plantee en clase.

El examen final, en el que se evaluarán los conocimientos de todo el curso, estará diseñado en base a preguntas cortas, largas y/o tipo test de elección múltiple (éste último con un nº mínimo de 20 preguntas). Para superar el examen, el alumno tendrá que alcanzar como mínimo el 50% de la puntuación de la parte teórica, y el 50% de la puntuación de la parte práctica.

Criterios de calificación:

La calificación final de la asignatura se expresará numéricamente, de acuerdo a lo dispuesto en el art. 5 del Real Decreto 1125/2003, de 5 de septiembre (BOE 18 de septiembre), por el que se establece el Sistema Europeo de Créditos y el Sistema de Calificaciones en las titulaciones universitarias de carácter oficial y su validez en todo el territorio nacional.

Sistema de Calificaciones:

0.0 - 4.9 Suspenso

5.0 - 6.9 Aprobado

7.0 - 8.9 Notable

9.0 - 10 Sobresaliente

Si el alumno no se presenta al examen en cualquiera de las convocatorias, se considerará \\\"No Presentado\\\". En caso de presentarse al examen, la calificación final será:

1. En la convocatoria ordinaria, $F = 0.6 \cdot E + 0.3 \cdot T + P$

2. En las convocatorias extraordinarias, $F = E$

donde,

E: La nota del examen, valorado en una escala de 0 a 10.

T: La nota media de los trabajos, valorados en una escala de 0 a 10.

P: Representa la asistencia y participación en clase en una escala de 0 a 10.

Para aprobar la asignatura F deberá ser mayor o igual que 5.

En una convocatoria ordinaria, un alumno que obtenga una calificación superior a 9.5 en el examen y en el trabajo, asiste a más del 90% todas las clases y participa activamente en ellas, es candidato a obtener una Matricula de Honor, que se asignará según la normativa vigente. En caso de que hayan más alumnos candidatos que Matriculas de Honor asignables, éstas se otorgarán por estricto orden de calificación final.

Plan de Aprendizaje (Plan de trabajo de cada estudiante)

Tareas y actividades que realizará según distintos contextos profesionales (científico, profesional, institucional, social)

La ingeniería del software ayudará a potenciar habilidades para:

1. Trabajar como parte de un equipo en el desarrollo y evolución de productos de software.
2. Comprender, aplicar y comunicar el proceso para determinar las necesidades del cliente y traducirlos a requisitos de software.
3. Conciliar objetivos en conflicto, considerando compromisos con las limitaciones de costo, tiempo, conocimiento, sistemas existentes y de las organizaciones involucradas.
4. Diseñar soluciones apropiadas en una o más áreas de concentración, usando enfoques de ingeniería que integren temas éticos, sociales, legales y económicos.
5. Entender y aplicar teorías, modelos y técnicas que provean una base para el diseño, desarrollo, verificación e implantación del software.
6. Negociar y trabajar profesionalmente, liderar cuando sea necesario y comunicarse efectivamente con los interesados en un ambiente de Ingeniería en Software.
7. Investigación en áreas relacionadas con la Ingeniería en Software.

Así mismo, se pretende que el alumno desarrolle actitudes de:

1. Liderazgo en equipos de trabajo multidisciplinarios.
2. Perseverancia en la solución de problemas.
3. Capacidad de mantenerse actualizado en su área de trabajo.
4. Afán de superación.
5. Responsabilidad y ética en su desempeño profesional.
6. Conducta emprendedora e innovadora.
7. Conciencia de la realidad social y responsabilidad ecológica.
8. Aprendizaje autodidacta.

Temporalización semanal de tareas y actividades (distribución de tiempos en distintas actividades y en presencialidad - no presencialidad)

Teórico (presencial)

1. Estructura del software (1 semanas)
2. Descomposición modular (3 semanas)
3. Dependencias (3 semanas)
4. Responsabilidades (3 semanas)
5. Abstracción (3 semanas)
6. Patrones de diseño (2 semanas)

Práctico (presencial)

1. Gestión de versiones del código (2 semanas)
2. Desarrollo del modelo (4 semanas)

3. Arquitectura Controlador, Vista, Modelo (3 semanas)
4. Principios y patrones de diseño (3 semanas)
5. Arquitectura Presenter (3 semanas)

Trabajo (no presencial)

1. Diseño de una aplicación con la arquitectura MVC
2. Diseño de una aplicación software con la arquitectura MVP

Recursos que tendrá que utilizar adecuadamente en cada uno de los contextos profesionales.

1. Entornos de desarrollo de software: Netbeans
2. Entorno de desarrollo Java Development Kit (JDK)
3. Sistemas de control de versiones: Git
4. Repositorios de código: GitHub, BitBucket.
5. Librerías Swing, AWT, SimpleXML

Resultados de aprendizaje que tendrá que alcanzar al finalizar las distintas tareas.

1. Trabajar como parte de un equipo en el desarrollo y evolución de productos de software.
2. Entender y aplicar el proceso para determinar las necesidades del cliente y traducirlos a requisitos de software.
3. Negociar objetivos en conflicto, considerando compromisos con las limitaciones de costo, tiempo, conocimiento, sistemas existentes y de las organizaciones involucradas.
4. Diseñar soluciones apropiadas en una o más áreas de concentración, usando enfoques de ingeniería.
5. Entender y aplicar teorías, modelos y técnicas que provean una base para desarrollo de software.
6. Negociar y trabajar profesionalmente, liderar cuando sea necesario y comunicarse efectivamente con los interesados en un ambiente de Ingeniería en Software.

Plan Tutorial

Atención presencial individualizada (incluir las acciones dirigidas a estudiantes en 5ª, 6ª y 7ª convocatoria)

En horario de tutoría previa cita del alumno a través de la plataforma Campus Virtual. En esta se habilitan las franjas horarias para asistir a tutorías, evitar solapes entre alumnos y así garantizar la atención individualizada un mínimo de 15 minutos.

Atención presencial a grupos de trabajo

En clases de laboratorio y horarios de tutoría previa cita del grupo a través de Campus Virtual. En esta se habilitan las franjas horarias para asistir a tutorías, evitar solapes entre alumnos y así garantizar la atención individualizada un mínimo de 15 minutos.

A final del curso, fuera del horario de clases, se realizarán diversos ejercicios prácticos preparatorios para el examen.

Atención telefónica

En horario de tutoría a través de Skype previa cita del alumno a través de la plataforma Campus Virtual. En esta se habilitan las franjas horarias para asistir a tutorías, evitar solapes entre alumnos y así garantizar la atención individualizada un mínimo de 15 minutos.

La dirección de skype se facilitará a los alumnos a través de Campus Virtual

Atención virtual (on-line)

En horario de tutoría, a través de correo electrónico y Campus Virtual

Bibliografía

[1 Básico] Patrones de diseño: elementos de software orientado a objetos reutilizable /

Erich Gamma... [et al.].

Addison Wesley,, Madrid : (2003)

9788478290598

[2 Básico] UML y patrones: introducción al análisis y diseño orientado a objetos /

Larman Craig.

Prentice Hall,, México : (1999)

970-17-0261-1

[3 Básico] Ingeniería del software: un enfoque práctico /

Roger S. Pressman.

McGraw Hill,, México [etc.] : (2010) - (7ª ed.)

9786071503145

[4 Recomendado] UML: el lenguaje unificado de modelado /

Grady Booch, James Rumbaugh, Ivar Jacobson.

Addison Wesley,, Madrid : (1999)

84-7829-028-1

[5 Recomendado] Software engineering /

Ian Sommerville.

Addison-Wesley,, Boston [etc.] : (2004) - (7th ed.)

0-321-21026-3

[6 Recomendado] Patrones de diseño aplicados a Java /

Stephen Stelting, Olav Maassen.

Prentice Hall,, Madrid : (2003)

8420538396