

CENTRO: 180 - Escuela de Ingeniería Informática

TITULACIÓN: 4008 - Grado en Ingeniería Informática

ASIGNATURA: 40975 - INGENIERÍA DEL SOFTWARE II

CÓDIGO UNESCO: **TIPO:** Obligatoria **CURSO:** 3 **SEMESTRE:** 1º semestre

CRÉDITOS ECTS: 6 **Especificar créditos de cada lengua:** **ESPAÑOL:** 6 **INGLÉS:**

SUMMARY

Software design is the process of defining the architecture of a software that comprises: components, interfaces and dependencies. This course is focused to understand design fundamentals: modularity, cohesion, coupling and abstraction. It includes also key design principles; design patterns and architecture styles; notations; strategies and design tools, all within the context of developing several applications.

REQUISITOS PREVIOS

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

Plan de Enseñanza (Plan de trabajo del profesorado)

Contribución de la asignatura al perfil profesional:

- C1. Capacidad para hacer diseño y programación orientada a objetos
- C2. Capacidad de un alto grado de reutilización del código
- C3. Capacidad para la construcción de librerías
- C4. Habilidades para utilizar herramientas CASE (UML)
- C5. Capacidad para aplicar los fundamentos y los principios del diseño, seleccionar patrones de diseño para resolver un problema concreto.
- C6. Capacidad para plantear arquitecturas flexibles orientadas al cambio constante.

Competencias que tiene asignadas:

- CB3. Que los estudiantes tengan la capacidad de 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.
- CB4. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- CB5. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

- G1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la Ingeniería en Informática que tengan por objeto la concepción, el desarrollo o la

explotación de sistemas, servicios y aplicaciones informáticas.

G3. 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.

G4. Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.

G5. 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.

G6. Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.

G8. 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.

G9. 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.

G10. Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

G11. 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.

G12. Conocimiento y aplicación de elementos básicos de economía y de gestión de recursos humanos, organización y planificación de proyectos, así como la legislación, regulación y normalización en el ámbito de los proyectos informáticos.

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.

N5. Participar activamente en la integración multicultural que favorezca el pleno desarrollo humano, la convivencia y la justicia social.

CI1. 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.

CI2. 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.

CI5. Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas.

CI6. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

CI7. Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados 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.

CI10. Conocimiento de las características, funcionalidades y estructura de los sistemas operativos y diseñar e implementar aplicaciones basadas en sus servicios.

CI11. Conocimiento y aplicación de las características, funcionalidades y estructura de los sistemas distribuidos, las redes de computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.

CI12. Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos.

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 del software.

CI17. Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

Objetivos:

Ob1. Mejorar la productividad y calidad en el desarrollo de software

Ob2. Conseguir que el estudiante adquiriera las capacidades para abordar un proceso de desarrollo de software

Ob3. Aprender herramientas para realizar Desarrollo dirigido por la prueba

Ob4. Aprender a diseñar software orientado a objetos

Ob5. Conseguir que el estudiante adquiriera las capacidades para abordar un diseño de software y su posterior implementación

Ob6. Aprender a describir el diseño del software usando métodos y técnicas de representación

Ob7. Aprender a refactorizar el código cuando se producen cambios en los requisitos

Ob8. Aprender a usar los estándares y patrones de diseño

Ob9. Aprender a gestionar las versiones del código

Contenidos:

PARTE TEORICA (30 horas: 2 horas/semana x 15 semanas)

1. Estructura del software
 - Definición y evolución
 - Programación estructurada
 - Estructura modular
 - Arquitecturas de software
 - Estilos de software
 - Patrones de diseño

Bibliografía. [PRESMAN]

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 4

2. Descomposición modular

- Modularidad
- Tipos de módulos
- Interfaz e implementación
- Cohesión y dependencia
- Jerarquía de control

Bibliografía. [PRESMAN] [CRAIG]

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 2

3. Dependencias

- Malos diseños
- Inyección de dependencias
- Inversión de dependencias
- Inversion del control

Bibliografía. [PRESMAN][CRAIG]

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 2

4. Responsabilidades

- Reglas de diseño
- Responsabilidad única
- Principio abierto-cerrado

Bibliografía. [PRESMAN][CRAIG]

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 2

5. Abstracción

- Formación de conceptos
- Generalización
- Sustituibilidad
- Principio de sustitución de Liskov
- Diseño por contrato

Bibliografía. [PRESMAN][CRAIG]

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 4

6. Desarrollo dirigido por la prueba (TDD y ATDD)

- Fundamentos
- Metodología
- Mocks y Stubs
- Test unitarios
- Test de integración
- Test de aceptación

Bibliografía: [C. BLE]

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 8

7. Patrones de diseño

- Patrones creacionales
- Patrones de estructura
- Patrones de comportamiento

Bibliografía. [GAMMA]

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 8

PARTE PRÁCTICA (30 horas: 2 horas/semana x 15 semanas)

1. Gestión de versiones del código

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 4

2. Desarrollo del modelo

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 8

3. Pruebas Unitarias y de Integración

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 6

4. Desarrollo de componentes del software guiado por Principios y Patrones de Diseño

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 6

5. Estilos Arquitectónicos, MVC y MVP

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 6

TRABAJO (no presencial) (30 horas)

1. Diseño de una aplicación software con la arquitectura MVP

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 10

2.- Diseño de pruebas unitarias y de integración en el software

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 10

3.- Diseño de una librería de Servicios Web sobre un conjunto de recursos (métodos a los que se puede llamar para actuar sobre esos recursos)

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Horas estimadas: 10

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 del alumnado 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 formativas que se realizarán a lo largo del curso en el ámbito de la asignatura y en función de su modalidad (presencial y no presencial).

Las actividades formativas presenciales que se emplearán serán las siguientes:

AF1.- Sesiones académicas teóricas en aula: exposición de los contenidos de la asignatura que capacitan al estudiante para la realización de las actividades prácticas.

AF2.- Sesiones académicas prácticas en aula: Se resolverán en el aula diferentes ejercicios propuestos para consolidar los conocimientos adquiridos en las sesiones teóricas.

AF3.- Sesiones académicas prácticas en laboratorio: desarrollo de determinadas prácticas utilizando los conocimientos adquiridos durante las sesiones teóricas.

AF4.- Tutorías individuales o en grupo: Se resolverán todas aquellas cuestiones planteadas por el alumno de manera individual o en grupo

Evaluación:

Criterios de evaluación

FE1.Examen escrito final

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

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Actividades: A1

FE2.Ejercicios y trabajos

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

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Actividades: A2

FE3.Asistencia a clase

La asistencia a clase contribuye a definir la nota de la evaluación continua. El control de asistencia incluye además la posibilidad de que el alumno/a responda a preguntas relacionadas con la clase de ese día. Estos controles se podrán realizar por escrito o mediante la plataforma del campus virtual.

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Actividades: A3, A4

FE4.Participación

La participación en clase la designa el profesor en función de la contribución que haya tenido en la clase con sus preguntas, comentarios o resolución de ejercicios que plantee el profesor.

Competencias: CB3, CB4, CB5, G1, G3, G4, G5, G6, G8, G9, G10, G11, G12, N1, N2, N3, N4, N5, CI1, CI2, CI5, CI6, CI7, CI8, CI10, CI11, CI12, CI13, CI16 y CI17.

Actividades: A1, A2

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 o la alumna, así como un examen al final.
- Para las pruebas extraordinaria y especial se realizará un único examen.

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/a 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 o la alumna 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 o la alumna 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, $F = 0.6 \cdot E + 0.3 \cdot T + 0.1 \cdot P$

2. En las convocatorias extraordinaria y especial, $F = 0.7 \cdot E + 0.3 \cdot T$

donde,

E: La nota del examen (FE1), valorado en una escala de 0 a 10.

T: La nota media de los trabajos (FE2), valorados en una escala de 0 a 10.

P: Representa la asistencia (FE3) y participación (FE4) en clase en una escala de 0 a 10. Proporcional al número de clases que asista.

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

En una convocatoria ordinaria, el alumno o la alumna 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 para obtener una Matrícula de Honor, que se asignará según la normativa vigente. En caso de que hayan más alumnos/as candidatos que Matrículas de Honor asignables, éstas se otorgarán por estricto orden de calificación final.

El estudiante tendrá la calificación No Presentado en el caso de que no entregue los trabajos (FE2) o no realice el examen (FE1)

Plan de Aprendizaje (Plan de trabajo de cada estudiante)

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

Las tareas y actividades que realizarán los estudiantes son:

Ta1. Adquirir conocimientos teóricos y prácticos sobre la estructura del software

Ta2. Adquirir conocimientos teóricos y prácticos sobre la descomposición modular, las dependencias y las responsabilidades del software

Ta3. Adquirir conocimientos teóricos y prácticos sobre la Abstracción en el desarrollo del software

Ta4. Adquirir conocimientos teóricos y prácticos sobre el desarrollo dirigido por la prueba

Ta5. Adquirir conocimientos teóricos y prácticos sobre el uso adecuado de los patrones de diseño en el desarrollo del software

Ta6. Adquirir conocimientos teóricos y prácticos sobre la gestión de versiones del código

Ta7. Adquirir conocimientos teóricos y prácticos sobre el desarrollo del modelo de un software

Ta8. Adquirir conocimientos teóricos y prácticos sobre cómo hacer una arquitectura Presenter en el desarrollo del software

Ta9. Adquirir conocimientos teóricos y prácticos sobre cómo identificar componentes del software guiado por Principios y Patrones de Diseño

Ta10. Adquirir conocimientos teóricos y prácticos sobre el desarrollo de Servicios Web sobre un conjunto de recursos

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

Semana 01 : Actividad 1 y 6: Estructura del Software. Gestión de versiones del código.

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 1 horas.

Semana 02 : Actividad 1 y 6: Estructura del Software. Gestión de versiones del código.

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 1 horas.

Semana 03 : Actividad 2 y 7: Descomposición modular, Dependencias y Responsabilidades. Desarrollo del modelo

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 2 horas.

Semana 04 : Actividad 2 y 7: Descomposición modular, Dependencias y Responsabilidades. Desarrollo del modelo

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 2 horas.

Semana 05 : Actividad 2 y 7: Descomposición modular, Dependencias y Responsabilidades. Desarrollo del modelo.

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 1 horas.

Semana 06 : Actividad 3 y 7: Abstracción. Desarrollo del modelo.

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 1 horas.

Semana 07 : Actividad 3 y 8: Abstracción. Identificación y desarrollo de Pruebas Unitarias e Integración.

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 3 horas.

Semana 08 : Actividad 4 y 8: Desarrollo dirigido por la prueba. Identificación y desarrollo de Pruebas Unitarias e Integración.

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 3 horas.

Semana 09 : Actividad 4 y 8: Desarrollo dirigido por la prueba. Identificación y desarrollo de Pruebas Unitarias e Integración.

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 3 horas.

Semana 10 : Actividad 4 y 9: Desarrollo dirigido por la prueba. Identificar componentes del software guiado por Principios y Patrones de Diseño

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 2 horas.

Semana 11 : Actividad 4 y 9: Desarrollo dirigido por la prueba. Identificar componentes del software guiado por Principios y Patrones de Diseño

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 2 horas.

Semana 12 : Actividad 5 y 9: Patrones de diseño y estilos arquitectónico. Identificar componentes del software guiado por Principios y Patrones de Diseño

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 3 horas.

Semana 13 : Actividad 5 y 10: Patrones de diseño y estilos arquitectónico MVC y MVP

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 2 horas.

Semana 14 : Actividad 5 y 10: Patrones de diseño y estilos arquitectónico MVC y MVP

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 2 horas.

Semana 15 : Actividad 5 y 10: Patrones de diseño y estilos arquitectónico MVC y MVP

Teoría: 2 horas. Prácticas: 2 horas. Trabajo no presencial: 2 horas.

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

Re1. Entornos de desarrollo de software: Netbeans

Re2. Entorno de desarrollo Java Development Kit (JDK)

Re3. Sistemas de control de versiones: Git

Re4. Repositorios de código: GitHub, BitBucket.

Re5. JUnit, Mockito

Re6. Librerías Swing, AWT, SimpleXML

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

RA1. Describir metodologías, principios, patrones y estilos de diseño de software.

Actividades: AF1, AF2, AF3, AF4

RA2. Aplicar teorías, modelos y técnicas fundamentales para el diseño y desarrollo de software usando enfoques de ingeniería.

Actividades: AF1, AF2, AF3, AF4

Plan Tutorial

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

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

Se diseñará un plan tutorial adaptado a cada alumno/a que estén en 5ª, 6ª o 7ª convocatoria, al cual se le realizará un seguimiento semanal.

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

Atención virtual (on-line)

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

Datos identificativos del profesorado que la imparte.

Datos identificativos del profesorado que la imparte

Dr./Dra. José Juan Hernández Cabrera (COORDINADOR)
Departamento: 260 - *INFORMÁTICA Y SISTEMAS*
Ámbito: 075 - *Ciencia De La Comp. E Intel. Artificial*
Área: 075 - *Ciencia De La Comp. E Intel. Artificial*
Despacho: *INFORMÁTICA Y SISTEMAS*
Teléfono: 928458752 **Correo Electrónico:** josejuan.hernandez@ulpgc.es

D/Dña. Octavio Roncal Andrés
Departamento: 260 - *INFORMÁTICA Y SISTEMAS*
Ámbito: 075 - *Ciencia De La Comp. E Intel. Artificial*
Área: 075 - *Ciencia De La Comp. E Intel. Artificial*
Despacho: *INFORMÁTICA Y SISTEMAS*
Teléfono: 928453301 **Correo Electrónico:** octavio.roncal@ulpgc.es

Bibliografía

[1 Básico] Diseño ágil con TDD /

Carlos Blé Jurado ... [et al.] ; prólogo de José Manuel Beas ; revisores Dácil Casanova...[et al.].
Carlos Blé Jurado,, [s.l.] : (2010)
978-1-4452-6471-4

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

Erich Gamma... [et al.].
Addison Wesley,, Madrid : (2003)
9788478290598

[3 Básico] UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado /

Larman Craig.
Prentice Hall,, Madrid : (2003) - (2ª ed.)
8420534382

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

Roger S. Pressman.
McGraw Hill,, México [etc.] : (2010) - (7ª ed.)
9786071503145

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

Grady Booch, James Rumbaugh, Ivar Jacobson.
Addison Wesley,, Madrid : (1999)
84-7829-028-1

[6 Recomendado] Software engineering /

Ian Sommerville.
Addison-Wesley,, Boston [etc.] : (2004) - (7th ed.)
0-321-21026-3

[7 Recomendado] Patrones de diseño aplicados a Java /

Stephen Stelting, Olav Maassen.
Prentice Hall,, Madrid : (2003)
8420538396