



Universidad
Internacional
de Valencia

Guía didáctica

ASIGNATURA: *Desarrollo del Software*

Título: *Grado en Ingeniería Informática*

Materia: *Mención en Ingeniería del Software*

Créditos: 6 ECTS

Código: 40GIIN

Índice

1. Organización general.....	3
1.1. Datos de la asignatura.....	3
1.2. Equipo docente	3
1.3. Introducción a la asignatura.....	3
1.4. Competencias y resultados de aprendizaje	4
2. Contenidos/temario	5
3. Metodología	6
4. Actividades formativas	6
5. Evaluación.....	7
5.1. Sistema de evaluación.....	7
5.2. Sistema de calificación	8
6. Bibliografía.....	9
6.1. Bibliografía de referencia	9
6.2. Bibliografía complementaria.....	9

1. Organización general

1.1. Datos de la asignatura

MÓDULO	Menciones
MATERIA	Mención en Ingeniería del Software
ASIGNATURA	<i>Desarrollo del Software</i> 6 ECTS
Carácter	Optativa
Curso	Tercero
Cuatrimestre	Segundo
Idioma en que se imparte	Castellano
Requisitos previos	Se recomienda haber cursado y superado las asignaturas de formación básica y obligatorias
Dedicación al estudio por ECTS	25 horas

1.2. Equipo docente

Profesor	María Paula Herrero <i>mariapaula.herrero@campusviu.es</i>
-----------------	--

1.3. Introducción a la asignatura

A través de esta asignatura se pretende hacer una introducción al entorno profesional del desarrollo de software que permita al alumno incorporarse a un proyecto conociendo de antemano el contexto en el que se desarrolla. Al final de este curso el alumno debe de haber asimilado y ser capaz de manejar los conceptos del desarrollo del software.

1.4. Competencias y resultados de aprendizaje

COMPETENCIAS GENERALES

- CG.1.- 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 la resolución, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
- CG.2.- Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.
- CG.3.- 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.
- CG.4.- Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.
- CG.5.- 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 la resolución.
- CG.7.- Capacidad para conocer, comprender y aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero Técnico en Informática y manejar especificaciones, reglamentos y normas de obligado cumplimiento.
- CG.9.- 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.
- CG.10.- Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en la resolución.
- CG.11.- 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.

COMPETENCIAS ESPECÍFICAS DE LA ASIGNATURA

- IS.1.- 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.
- IS.3.- Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

- IS.4.- 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.
- IS.6.- Capacidad para diseñar soluciones apropiadas en uno o más dominios de aplicación utilizando métodos de la ingeniería del software que integren aspectos éticos, sociales, legales y económicos.

RESULTADOS DE APRENDIZAJE

Al finalizar esta asignatura se espera que el estudiante sea capaz de:

- RA.1.- Identificar los patrones que se pueden aplicar a un diseño, su estructura y clasificación y su importancia como herramienta para mejorar la calidad de un diseño.
- RA.2.- Explicar las técnicas de desarrollo dirigidas por modelos, sus ventajas e inconvenientes.
- RA.3.- Definir pruebas unitarias de integración, regresión y sistema, e implementarlas.
- RA.4.- Determinar la importancia del mantenimiento y su integración en el proceso de desarrollo de software y evaluar el coste del mantenimiento de un sistema.

2. Contenidos/temario

Unidad Competencial 1 /Tema 1. Desarrollo utilizando patrones-software

- 1.1. Ingeniería del software
- 1.2. Proyectos de software
- 1.3. Características del software
- 1.4. Proceso de creación de un producto de software
- 1.5. El ciclo de vida de un producto de software
- 1.6. Modelos de ciclo de vida tradicionales
- 1.7. Modelos de ciclo de ágil

Unidad Competencial 2 /Tema 2. Desarrollo dirigido por modelos

- 2.1. Diagramas UML
- 2.2. Modelo del Dominio
- 3.3. Requisitos

Unidad Competencial 3 /Tema 3. Arquitecturas Software

- 3.1. Definición de arquitectura del software
- 3.2. Patrones de arquitectura de software
- 3.3. Diseño de software

Unidad Competencial 4 /Tema 4. Pruebas de software

- 4.1. Casos de test
- 4.2. Tipos de pruebas
- 4.3. Herramientas

Unidad Competencial 5 /Tema 5. Mantenimiento y evolución del software

- 5.1. Planes de Entrega
- 5.2. Mantenimiento
- 5.3. Integración Continua
- 5.4. Refactorización

3. Metodología

La metodología de la Universidad Internacional de Valencia (VIU) se caracteriza por una apuesta decidida en un modelo de carácter e-presencial. Así, siguiendo lo estipulado en el calendario de actividades docentes del Título, se impartirán en directo un conjunto de sesiones, que, además, quedarán grabadas para su posterior visionado por parte de aquellos estudiantes que lo necesiten. En todo caso, se recomienda acudir, en la medida de lo posible, a dichas sesiones, facilitando así el intercambio de experiencias y dudas con el docente.

En lo que se refiere a las metodologías específicas de enseñanza-aprendizaje, serán aplicadas por el docente en función de los contenidos de la asignatura y de las necesidades pedagógicas de los estudiantes. De manera general, se impartirán contenidos teóricos y, en el ámbito de las clases prácticas se podrá realizar la resolución de problemas, el estudio de casos y/o la simulación.

Por otro lado, la Universidad y sus docentes ofrecen un acompañamiento continuo al estudiante, poniendo a su disposición foros de dudas y tutorías para resolver las consultas de carácter académico que el estudiante pueda tener. Es importante señalar que resulta fundamental el trabajo autónomo del estudiante para lograr una adecuada consecución de los objetivos formativos previstos para la asignatura.

4. Actividades formativas

Durante el desarrollo de cada una de las asignaturas se programan una serie de actividades de aprendizaje que ayudan a los estudiantes a consolidar los conocimientos trabajados.

A continuación, se relacionan las actividades que forman parte de la asignatura:

1. Actividades de carácter teórico

Se trata de un conjunto de actividades guiadas por el profesor de la asignatura destinadas a la adquisición por parte de los estudiantes de los contenidos teóricos de la misma. Estas actividades, diseñadas de manera integral, se complementan entre sí y están directamente relacionadas con los materiales teóricos que se ponen a disposición del estudiante (manual, SCORM y material complementario). Estas actividades se desglosan en las siguientes categorías:

- a. Clases expositivas
- b. Sesiones con expertos en el aula

- c. Observación y evaluación de recursos didácticos audiovisuales
- d. Estudio y seguimiento de material interactivo

2. Actividades de carácter práctico

Se trata de un conjunto de actividades guiadas y supervisadas por el profesor de la asignatura vinculadas con la adquisición por parte de los estudiantes de los resultados de aprendizaje y competencias de carácter más práctico. Estas actividades, diseñadas con visión de conjunto, están relacionadas entre sí para ofrecer al estudiante una formación completa e integral.

3. Tutorías

Se trata de sesiones, tanto de carácter síncrono como asíncrono (e-mail), individuales o colectivas, en las que el profesor comparte información sobre el progreso académico del estudiante y en las que se resuelven dudas y se dan orientaciones específicas ante dificultades concretas en el desarrollo de la asignatura.

4. Trabajo autónomo

Se trata de un conjunto de actividades que el estudiante desarrolla autónomamente y que están enfocadas a lograr un aprendizaje significativo y a superar la evaluación de la asignatura. La realización de estas actividades es indispensable para adquirir las competencias y se encuentran entroncadas en el aprendizaje autónomo que consagra la actual ordenación de enseñanzas universitarias. Esta actividad, por su definición, tiene carácter asíncrono.

5. Prueba objetiva final

Como parte de la evaluación de cada una de las asignaturas (a excepción de las prácticas y el Trabajo fin de título), se realiza una prueba (examen final). Esta prueba se realiza en tiempo real (con los medios de control antifraude especificados) y tiene como objetivo evidenciar el nivel de adquisición de conocimientos y desarrollo de competencias por parte de los estudiantes. Esta actividad, por su definición, tiene carácter síncrono.

5. Evaluación

5.1. Sistema de evaluación

El Modelo de Evaluación de estudiantes en la Universidad se sustenta en los principios del Espacio Europeo de Educación Superior (EEES), y está adaptado a la estructura de formación virtual propia de esta Universidad. De este modo, se dirige a la evaluación de competencias.

Sistema de Evaluación	Ponderación
Portafolio*	50 %
<i>Entrega de informes de problemas y ejercicios</i>	20%
<i>Planteamiento, estudio, análisis y resolución de casos</i>	10%
<i>Informes o memorias de prácticas de laboratorio</i>	15%
<i>Participación activa en los debates, foros y otros medios</i>	5%
Sistema de Evaluación	Ponderación
Prueba final*	50 %
<i>La prueba final consistirá en un examen en línea, el cual incluirá preguntas de tipo teóricas, así como resolución de ejercicios y/o problemas. Esto permitirá que el estudiante demuestre que ha alcanzado las competencias y resultados de aprendizaje esperados en cada una de las unidades competenciales.</i>	

***Es requisito indispensable para superar la asignatura aprobar cada apartado (portafolio y prueba final) con un mínimo de 5 para ponderar las calificaciones.**

Los enunciados y especificaciones propias de las distintas actividades serán aportados por el docente, a través del Campus Virtual, a lo largo de la impartición de la asignatura.

Atendiendo a la Normativa de Evaluación de la Universidad, se tendrá en cuenta que la utilización de **contenido de autoría ajena** al propio estudiante debe ser citada adecuadamente en los trabajos entregados. Los casos de plagio serán sancionados con suspenso (0) de la actividad en la que se detecte. Asimismo, el uso de **medios fraudulentos durante las pruebas de evaluación** implicará un suspenso (0) y podrá implicar la apertura de un expediente disciplinario.

5.2. Sistema de calificación

La calificación de la asignatura se establecerá en los siguientes cálculos y términos:

Nivel de aprendizaje	Calificación numérica	Calificación cualitativa
Muy competente	9,0 - 10	Sobresaliente
Competente	7,0 - 8,9	Notable
Aceptable	5,0 -6,9	Aprobado
Aún no competente	0,0 -4,9	Suspenso

Sin detrimento de lo anterior, el estudiante dispondrá de una **rúbrica simplificada** en el aula que mostrará los aspectos que valorará el docente, como así también los **niveles de desempeño que tendrá en cuenta para calificar las actividades vinculadas a cada resultado de aprendizaje**.

La mención de «**Matrícula de Honor**» podrá ser otorgada a estudiantes que hayan obtenido una calificación igual o superior a 9.0. Su número no podrá exceder del cinco por ciento de los estudiantes matriculados en una materia en el correspondiente curso académico, salvo que el número de estudiantes matriculados sea inferior a 20, en cuyo caso se podrá conceder una sola «Matrícula de Honor».

6. Bibliografía

6.1. Bibliografía de referencia

- IEEE Standard for System and Software Verification and Validation, in IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004), vol., no., pp.1-223, 25 May 2012 doi: 10.1109/IEEESTD.2012.6204026
<https://ieeexplore.ieee.org/document/6204026>
- Schmidt, Richard and Schmidt Richard F. (2013). Software Engineering: Architecture-Driven Software Development. Elsevier Science & Technology.
<https://ebookcentral.proquest.com/lib/universidadviu/detail.action?docID=1187150>
- Babar, Muhammad Ali, Alan W. Brown, and Ivan Mistrik (2013). Agile Software Architecture: Aligning Agile Processes and Software Architectures. Elsevier Science & Technology.
<https://ebookcentral.proquest.com/lib/universidadviu/reader.action?docID=1574922>
- Lano, Kevin (2009). UML 2 Semantics and Applications. John Wiley & Sons, Incorporated.
<https://ebookcentral.proquest.com/lib/universidadviu/reader.action?docID=711732>
- Myers, Glenford J., Corey Sandler, and Tom Badgett (2011). The Art of Software Testing. John Wiley & Sons, Incorporated.
<https://ebookcentral.proquest.com/lib/universidadviu/reader.action?docID=697721>
- Schwaber, Ken y Jeff Sutherland (2020). La Guía Definitiva de Scrum: Las Reglas del Juego. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

6.2. Bibliografía complementaria

- Anderson, D. J. (2010). Kanban: successful evolutionary change for your technology business. Blue Hole Press. ISBN 0-9845214-0-2
- Garlan, David, and Mary Shaw (Enero, 1994). An Introduction to Software Architecture. Carnegie Mellon University Technical Report, CMU-CS-94-166.

- Martin, R. C. (2002). Agile software development: principles, patterns, and practices. Prentice Hall.
- Jacobson, Ivar, Grady Booch y James Rumbaugh (2000). El Proceso Unificado de Desarrollo de Software. Addison-Wesley.
- Kniberg, Henrik (2015). Scrum and XP from the Trenches. Info Q, Enterprise Software Development Series.
- Meszaros, Gerard (2007). xUnit Test Patterns. Refactoring Test Code. Addison-Wesley.