



Universidad
Internacional
de Valencia

Guía didáctica

ASIGNATURA: *Mantenimiento y Evolución del Software*

Título: *Grado en Ingeniería Informática*

Materia: *Mención en Ingeniería del Software*

Créditos: 6 ECTS

Código: 43GIIN

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.....	8
5.1.	Sistema de evaluación.....	8
5.2.	Sistema de calificación	9
6.	Bibliografía.....	9
6.1.	Bibliografía de referencia.....	9
6.2.	Bibliografía complementaria.....	11

1. Organización general

1.1. Datos de la asignatura

MÓDULO	Menciones
MATERIA	Mención en Ingeniería del Software
ASIGNATURA	<i>Mantenimiento y Evolución del Software</i> 6 ECTS
Carácter	Optativa
Curso	Tercero
Cuatrimestre	Segundo
Idioma en que se imparte	Castellano
Requisitos previos	No existen
Dedicación al estudio por ECTS	25 horas

1.2. Equipo docente

Profesor	Dr. Eduardo Zamudio <i>eduardo.zamudio@professor.universidadviu.com</i>
-----------------	---

1.3. Introducción a la asignatura

Esta asignatura pertenece tercer curso del Grado Ingeniería Informática Mención Ingeniería de Software.

En esta asignatura buscaremos comprender la importancia de concebir el mantenimiento como parte de un proceso de ingeniería de software.

El punto de partida de la asignatura se basa en garantizar el entendimiento de que el software es un producto que se encuentra inmerso en un ambiente cambiante. Los cambios producidos en el ambiente sugieren la necesidad de adaptar el producto a dichos cambios. Asimismo, el software es un producto humano y es susceptible de presentar defectos, los cuales deben ser corregidos para mantener el producto en operación.

Los contenidos de la asignatura incluyen desde aspectos conceptuales del proceso de mantenimiento, la definición de mantenibilidad y su relación con el ciclo de vida del producto software, hasta una descripción de algunos de los elementos destacados de la gestión del mantenimiento.

Los elementos relacionados con el mantenimiento del software aquí presentados incluyen la Gestión de Versiones y características de las herramientas de seguimientos de defectos, la

Pruebas y su importancia en los procesos de Verificación y Validación del software, elementos asociados a la calidad del software como las guías de estilo de código y su relación con la mantenibilidad, la reingeniería y la ingeniería inversa como estrategias de modificación de la arquitectura del software, así como la presentación de estándares relacionados con la mantenibilidad del software.

Objetivos generales

Los objetivos generales incluyen:

- *Comprender la importancia del mantenimiento del producto software*
- *Conocer los principales componentes del proceso de mantenimiento de software*
- *Desarrollar habilidades relacionadas con el uso de elementos asociados al mantenimiento de 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.

RESULTADOS DE APRENDIZAJE

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

- RA.1.- Describir los procesos y niveles de prueba desarrollados en proyectos de software.
- RA.2.- Aplicar procesos de revisión y auditoría a productos habituales de los proyectos de software y gestionarlos.
- RA.3.- Realizar evaluaciones y predicciones básicas de fiabilidad de software.
- RA.4.- Diseñar esquemas de medición de los productos y los procesos más significativos del ciclo de vida de software y aplicarlos.

2. Contenidos/temario

Tema 1 Introducción
Tema 2 Gestión de versiones
Tema 3 El proceso de mantenimiento y ciclo de vida
Tema 4 Herramientas de seguimiento de defectos
Tema 5 Ingeniería Inversa y Reingeniería
Tema 6 El Estándar ISO/IEC 14764
Tema 7 Pruebas
Tema 8 Guías de estilo de código Tema 9 Mantenibilidad

Agrupación por unidades

Unidad Competencial 1

Tema 1 Introducción
Tema 3 El proceso de mantenimiento y ciclo de vida
Tema 6 El Estándar ISO/IEC 14764

Unidad Competencial 2

Tema 2 Gestión de versiones
Tema 8 Guías de estilo de código

Unidad Competencial 3

Tema 7 Pruebas

Tema 4 Herramientas de seguimiento de defectos

Unidad Competencial 4

Tema 5 Ingeniería Inversa y Reingeniería

Tema 9 Mantenibilidad

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

Se prevén sesiones sincrónicas en las que el profesor realizará exposiciones con los contenidos teóricos y la presentación de casos prácticos. Asimismo, se usarán los espacios sincrónicos para fomentar el debate con los estudiantes.

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.

Los estudiantes tendrán la capacidad de decidir cuándo entregar las actividades del portafolio:

- En la fecha final propuesta para la entrega del portafolio en cada una de las convocatorias
- Días antes del examen (sin feedback)

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.

Se impartirán tutorías colectivas en forma síncrona mediante videoconferencias al inicio y al final de la asignatura. En la primera se presentará la asignatura (profesorado, planificación y material recomendado) y la segunda estará destinada a resolver las dudas planteadas por el alumnado, a su valoración sobre el desarrollo de la asignatura, y a la preparación de la evaluación. Estas clases deberán seguirse en el horario establecido en la planificación de la asignatura, que quedarán grabadas para un posible visionado posterior.

El estudiante podrá realizar consultas por correo electrónico. Existirá además la posibilidad de realizar tutorías individuales mediante sesiones de videoconferencia mediante petición previa vía e-mail, planificadas con suficiente antelación donde se buscará un horario adecuado tanto para el estudiante como para el profesor.

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.

Las actividades pueden ser desarrolladas en forma grupal o individual. En la opción grupal, todos los miembros deben conocer todas las actividades y ser capaces de demostrarlo durante el examen.

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>	10%
<i>Planteamiento, estudio, análisis y resolución de casos</i>	15%
<i>Informes o memorias de prácticas de laboratorio</i>	10%
<i>Trabajos o proyectos desarrollados en grupo o de forma individual</i>	10%
<i>Participación activa en los debates, foros y otros medios</i>	5%
Sistema de Evaluación	Ponderación
Prueba final*	50 %
<i>Prueba con componentes teóricos y prácticos sobre el temario de la asignatura y las actividades desarrolladas durante todo el cuatrimestre.</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

Apache Software Foundation. (n.d.). Apache Subversion. Retrieved May 10, 2019, from <https://subversion.apache.org/>

AXELOS. (n.d.). ITIL | IT Service Management | ITSM | AXELOS. Retrieved May 1, 2019, from <https://www.axelos.com/best-practice-solutions/itil>

Bourque, P., & Fairley, R. E. (2014). SWEBOK Guide V3.0. IEEE Computer Society. <https://doi.org/10.1234/12345678>

Bugzilla. (n.d.). Bugzilla Documentation — Bugzilla 5.1.2+ documentation. Retrieved July 14, 2019, from <https://bugzilla.readthedocs.io/en/latest/>

cachéQuality.(n.d.). cachéQuality. Retrieved July 17, 2019, from <https://www.cachequality.com/>

Chacon, S., & Straub, B. (2014). Pro Git (2nd ed.). Berkely, CA, USA: Apress.

Driessen, V. (2010). A successful Git branching model » nvie.com. Retrieved May 12, 2019, from <https://nvie.com/posts/a-successful-git-branching-model/>

Ernst, M. (2012). Version control concepts and best practices. Retrieved May 1, 2019, from <https://homes.cs.washington.edu/~mernst/advice/version-control.html>

Free Software Foundation. (n.d.-a). CVS - Open Source Version Control. Retrieved May 10, 2019, from <https://www.nongnu.org/cvs/>

Free Software Foundation. (n.d.-b). RCS. Retrieved May 10, 2019, from <http://www.gnu.org/software/rcs/rcs.html>

Git. (n.d.). git/git: Git Source Code Mirror. Retrieved May 12, 2019, from <https://github.com/git/git>

Google.(n.d.). Google Java Style Guide. Retrieved July 17, 2019, from <https://google.github.io/styleguide/javaguide.html>

ISACA.(n.d.-a). CMMI Institute - CMMI Services. Retrieved May 1, 2019, from <https://cmmiinstitute.com/cmmi/svc>

ISACA. (n.d.-b). COBIT 2019. Retrieved May 1, 2019, from <http://www.isaca.org/COBIT>

ISO/IEC. (n.d.-a). ISO/IEC 25010:2011(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Retrieved July 17, 2019, from <https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en>

ISO/IEC. (n.d.-b). ISO/IEC 20000-1:2018 - Information technology -- Service management -- Part 1: Service management system requirements. Retrieved May 1, 2019, from <https://www.iso.org/standard/70636.html>

ISO, IEC, & IEEE. (2006). ISO/IEC 14764:2006 - Software Engineering — Software Life Cycle Processes — Maintenance. Electronics (Vol. 2006). Retrieved from <https://standards.ieee.org/standard/23026-2006.html>

MDN. (n.d.). Bug report writing guidelines. Retrieved July 14, 2019, from https://developer.mozilla.org/en-US/docs/Mozilla/QA/Bug_writing_guidelines

Mercurial. (n.d.). Mercurial SCM. Retrieved May 10, 2019, from <https://www.mercurial-scm.org/about>

Microsoft. (n.d.). Code Metrics – Maintainability Index – The Ultimate Visual Studio Tips and Tricks Blog. Retrieved July 17, 2019, from <https://blogs.msdn.microsoft.com/zainnab/2011/05/26/code-metrics-maintainability-index/>

Miller, P. (n.d.). Aegis. Retrieved May 10, 2019, from <http://aegis.sourceforge.net/>

Oman, P., & Hagemester, J. (n.d.). Metrics for assessing a software system's maintainability. In Proceedings Conference on Software Maintenance 1992 (pp. 337–344). IEEE Comput. Soc. Press. <https://doi.org/10.1109/ICSM.1992.242525>

Oracle. (n.d.). Code Conventions for the Java Programming Language: Contents. Retrieved July 17, 2019, from <https://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

PMI. (n.d.). PMBOK Guide and Standards | Project Management Institute. Retrieved May 1, 2019, from <https://www.pmi.org/pmbok-guide-standards>

Pressman, R. S. (2010). Software Engineering : a practitioner's approach. McGraw Hill Education (India) Private Limited. Retrieved from <https://www.amazon.com/Software-Engineering-Practitioners-Approach-7TH/dp/B01N284CFH>

ProjectCodeMeter. (n.d.). ProjectCodeMeter - ProjectCodeMeter Software Sizing for Outsourcing Work Hours Assessment and Development Cost Estimation. Retrieved July 17, 2019, from http://www.projectcodemeter.com/cost_estimation/index.html

Python. (n.d.). PEP 8 -- Style Guide for Python Code | Python.org. Retrieved July 16, 2019, from <https://www.python.org/dev/peps/pep-0008/#id8>

Radon. (n.d.). Welcome to Radon's documentation! — Radon 2.4.0 documentation. Retrieved July 17, 2019, from <https://radon.readthedocs.io/en/latest/index.html>

Rosenberg, L. H., & Hyatt, L. E. (1997). Software Re-engineering. Retrieved from <https://pdfs.semanticscholar.org/d6b2/89019f9fef924fd3300d1094f29c8bc079f9.pdf>

Software Freedom Conservancy. (n.d.). Git. Retrieved May 10, 2019, from <https://git-scm.com/> Sommerville, I. (2016). Software engineering. Pearson.

Wijnholds, G., Van Eck, P., Van Der Leek, R., Rigal, S., & Visser, J. (2016). Building maintainable software : ten guidelines for future-proof code. O'Reilly Media Inc.

6.2. Bibliografía complementaria

Artículo comparativo de 15 herramientas CVS en la que se identifican pros y cons de cadauna. <https://www.softwaretestinghelp.com/version-control-software/>

Artículo sobre la historia del CVS previo a la aparición de Git. El blog además trata la historia de la Ciencia de la Computación en general.
<https://twobithistory.org/2018/07/07/cvs.html>

Artículo sobre el modelo de branching de Git.

<https://nvie.com/posts/a-successful-git-branching-model/>

Artículo sobre distintos modelos de mantenimiento de software

<http://www.professionalqa.com/software-maintenance-models>

Artículo sobre herramientas para la automatización de pruebas de software

<https://dzone.com/articles/top-10-automated-software-testing-tools>

Artículo de Wikipedia en el que se mencionan varias convenciones de nombres para distintos lenguajes. [https://en.wikipedia.org/wiki/Naming_convention_\(programming\)](https://en.wikipedia.org/wiki/Naming_convention_(programming))

Artículo de Wikipedia con un resumen de las lecciones a considerar de los estilos de programación. https://en.wikipedia.org/wiki/The_Elements_of_Programming_Style

Artículo crítico sobre el Índice de Mantenibilidad.

<https://avandeursen.com/2014/08/29/think-twice-before-using-the-maintainability-index/>