

CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL



**PLANIFICACIÓN DE LA DOCENCIA
UNIVERSITARIA**

GUÍA DOCENTE

Verificación de Software

1. DATOS DE IDENTIFICACIÓN DE LA ASIGNATURA.

Título:	Grado en Ingeniería del Software
Facultad:	Centro Universitario de Tecnología y Arte Digital (U-TAD)
Materia:	Optatividad
Denominación de la asignatura:	Verificación del SW
Curso:	4
Cuatrimestre:	2
Carácter:	Optativa
Créditos ECTS:	3
Modalidad/es de enseñanza:	Híbrido Presencial
Idioma:	Castellano
Profesor/a:	Alonso Álvarez Garcia
E-mail:	alonso.alvarez@ext.live.u-tad.com
Teléfono:	

2. DESCRIPCIÓN DE LA ASIGNATURA.

2.1 Descripción de la materia

Esta materia recoge algunos contenidos avanzados y/o especializados que pudiera requerir un ingeniero del software generalista.

2.2 Descripción de la asignatura

Esta asignatura es un acercamiento a la calidad del software. Por ello, se centra en entender los conceptos básicos en relación con la calidad y las características especiales cuando se aplica al mundo del software.

Se da especial relevancia al análisis de casos de uso, la planificación y seguimiento de las pruebas y, en general, a los elementos básicos presentes en cualquier proceso de testing de software, con independencia de la forma en la que se implemente.

Al mismo tiempo, la asignatura introducirá técnicas y prácticas habituales en el mundo de la calidad del software para que ofrecer una primera experiencia aplicable en el desempeño profesional. Esas técnicas y prácticas son las más frecuentes a la hora de diseñar y especificar casos de prueba, paso previo a su ejecución manual o automatizada de los mismos.

3. COMPETENCIAS

Competencias Básicas y Generales

CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio 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.

CB2: Que los estudiantes sepan 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.

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.

CG1: Entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas.

CG2: Desarrollar soluciones informáticas que sean respetuosas con el medio ambiente, los deberes sociales y recursos naturales, además de cumplir con la legislación y ética.

CG3: Aplicar los fundamentos científicos para la resolución de problemas informáticos

CG4: Entender la complejidad, simplificar y optimizar los sistemas informáticos

CG6: Trabajar en entornos de trabajo multidisciplinares demostrando capacidad de trabajo en equipo, versatilidad, flexibilidad, creatividad y respeto por el trabajo de los compañeros de otras áreas.

CG7: Aplicar los fundamentos creativos de generación de ideas en los proyectos de desarrollo software para entornos digitales.

CG9: Aprender, modificar y producir nuevas tecnologías informáticas

CG10: Aplicar las técnicas creativas para la realización de proyectos informáticos

CG11: Buscar, analizar y gestionar la información para poder extraer conocimiento de la misma.

Competencias Específicas

CE10: Generar documentación de una aplicación de forma automática así como entender y manejar adecuadamente un gestor de versiones de código las que utilizan principios de ingeniería inversa.

CE15: Desarrollar aplicaciones distribuidas teniendo en cuenta la tolerancia de los fallos, la adaptabilidad, el balance de carga y la predictividad del sistema.

CE17: Desarrollar aplicaciones que utilicen las características de paralelización de tarjetas gráficas y arquitecturas de altas prestaciones.

CE20: Testar en profundidad el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas.

CE21: Evaluar la calidad de una aplicación informática desde el punto de vista de su diseño e implementación, aplicando métricas, procedimientos y estándares de medición de calidad del software.

Competencias Transversales

CT1: Conocer la definición y el alcance, así como poner en práctica los fundamentos de las metodologías de gestión de proyectos de desarrollo tecnológico.

CT2: Conocer los principales agentes del sector y el ciclo de vida completo de un proyecto en desarrollo y comercialización de contenidos digitales

CT4: Actualizar el conocimiento adquirido en el manejo de herramientas y tecnologías digitales en función del estado actual del sector y de las tecnologías empleadas.

CT5: Poseer las habilidades necesarias para el emprendimiento digital.

4. CONTENIDOS

Tema 1. Fundamentos de Calidad del Software

- 1.1. Orígenes, Calidad en manufactura
- 1.2. Conceptos básicos: QA, QC, testing
- 1.3. Verificación del software o testing
- 1.4. Los siete principios del testing
- 1.5. La mentalidad del testing

Tema 2. QA en el SDLC

- 2.1. Calidad del software en desarrollo predictivo
- 2.2. Una aproximación formal a la Calidad en proyectos: modelo PMP
- 2.3. Calidad del software en paradigma ágil
- 2.4. Calidad en frameworks Agile
- 2.5. Mantenimiento y soporte

Tema 3. Revisión y pruebas

- 3.1. Tipos de pruebas
- 3.2. Análisis estático de código
- 3.3. Revisión
- 3.4. Pruebas de caja negra
- 3.5. Pruebas de caja blanca
- 3.6. Pruebas basadas en experiencia

Tema 4. Técnicas y herramientas

- 4.1. Modelos orientados a pruebas: TDD, BDD, ATDD
- 4.2. Automatización de pruebas
- 4.3. CI/CD y DevOps

Tema 5. Gestión de las pruebas

- 5.1. Herramientas
- 5.2. Reporting y control. Trazabilidad
- 5.3. Escenarios de despliegue y prueba

4.1. Desarrollo temporal

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
Tema 1. Fundamentos de Calidad del Software	3 semanas
Tema 2. QA en el SDLC	3 semanas
Tema 3. Revisión y pruebas	3 semanas
Tema 4. Técnicas y herramientas	4 semanas
Tema 5. Gestión de las pruebas	2 semanas

5. ACTIVIDADES FORMATIVAS Y MODALIDADES DE ENSEÑANZAS

5.1 Modalidades de enseñanza

La asignatura se desarrollará a través de los siguientes métodos y técnicas generales, que se aplicarán diferencialmente según las características propias de la asignatura:

- **Método expositivo/Lección magistral:** el profesor desarrollará, mediante clases magistrales y dinámicas los contenidos recogidos en el temario.
- **Estudio de casos:** análisis de casos reales relacionados con la asignatura.
- **Resolución de ejercicios y problemas:** los estudiantes desarrollarán las soluciones adecuadas aplicando procedimientos de transformación de la información disponible y la interpretación de los resultados.
- **Aprendizaje basado en problemas:** utilización de problemas como punto de partida para la adquisición de conocimientos nuevos.
- **Aprendizaje orientado a proyectos:** se pide a los alumnos que, en pequeños grupos, planifiquen, creen y evalúen un proyecto que responda a las necesidades planteadas en una determinada situación.
- **Aprendizaje cooperativo:** Los estudiantes trabajan en grupo para realizar las tareas de manera colectiva.

5.2 Actividades formativas

Actividad Formativa	Horas	Presencialidad
AF1 Clases teóricas / Expositivas	16	100%
AF2 Clases Prácticas	11	100%
AF3 Tutorías	3	50%
AF4 Estudio independiente y trabajo autónomo del alumno	25	0%
AF5 Elaboración de trabajos (en grupo o individuales)	18	0%
AF6: Actividades de Evaluación	2	100%

6. SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
SE1 Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	10%	30%
SE2 Evaluación de trabajos, proyectos, informes, memorias	30%	60%
SE3 Prueba Objetiva	30%	60%

6.1. Criterios de calificación

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN RESPECTO A LA CALIFICACIÓN FINAL (%)
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	10%
Evaluación de trabajos, proyectos, informes, memorias	30%
Prueba Objetiva	60%

- La evaluación de la participación en clase, en prácticas o en proyectos de la asignatura se realizará a partir de la asistencia y la participación activa en clase y en el resto de las actividades desarrolladas durante el curso. Este aspecto representará el 10% de la calificación final de la asignatura en la convocatoria ordinaria.
- A lo largo del curso se plantearán actividades, ejercicios y problemas que deberán ser entregadas antes de la fecha indicada a través de la plataforma virtual. Este trabajo se evaluará a través de la propia plataforma virtual y supondrá un 30% de la calificación final de la asignatura en la convocatoria ordinaria. No se admitirán trabajos fuera de forma y fecha sin causa justificada. En caso de que la nota de un ejercicio no alcance el aprobado, se admitirá una nueva entrega que si tiene la suficiente calidad tendrá una calificación de 5.0.
- Para aprobar la asignatura en la convocatoria ordinaria, es imprescindible que la nota final sea al menos 5.0 (sobre 10). Además de ese requisito, es necesario que la media de los trabajos planteados sea al menos 5.0 (sobre 10), y que la nota del

examen final sea al menos de 4.0 (sobre 10). En caso de no cumplirse alguno de estos requisitos, la asignatura se considerará automáticamente suspensa independientemente del resto de calificaciones.

- En caso de no conseguir el aprobado en la convocatoria ordinaria, el alumno podrá presentarse a la convocatoria extraordinaria. Se realizará un examen final que representará el 70% de su calificación en dicha convocatoria, y en el que formará parte de la materia exigible al alumno todo el contenido de la asignatura visto en clase. El otro 30% se calculará a partir de un ejercicio o ejercicios práctico entregable que será planteado al comienzo del comienzo de segundo cuatrimestre. Para aprobar la asignatura en la convocatoria extraordinaria, es imprescindible que la nota final sea al menos 5.0 (sobre 10). Además de ese requisito, es necesario que la media de los trabajos planteados sea al menos 5.0 (sobre 10), y que la nota del examen final sea al menos de 4.0 (sobre 10).
- En los exámenes no se permite el uso de apuntes ni de calculadoras científicas programables, para lo que el alumno debe remitirse a las instrucciones específicas del profesor sobre este tema.
- Todo el código y trabajos entregados por los alumnos deberán ser ORIGINALES. Quiere decir que deberán haber sido desarrollados por los alumnos a lo largo de la asignatura, sin ayuda externa. En caso de usar código/librerías externas a lo suministrado por el profesor, deberá estar debidamente documentado y justificado. Se permite consultar documentación externa a la asignatura, pero el código entregado por el alumno deberá respetar las leyes de copyright y licencias software vigentes. En todo caso, el alumno deberá ser capaz de explicar el código usado y entregado durante el curso.
- **Copias entre trabajos:** Se entenderá como copia de trabajo aquellos proyectos que contengan partes iguales o muy similares, que no cumplan las reglas establecidas en los párrafos anteriores. Las copias de trabajos conllevarán la completa suspensión de la asignatura, sin posibilidad de recuperación en la convocatoria actual. Será el profesor el que decida la gravedad de la copia, y la decisión final podrá ser consultada y revocada por el resto del equipo docente en caso de necesitar una segunda opinión.
- No se conservarán calificaciones de ningún tipo entre distintos cursos académicos, ni entre distintas convocatorias.
- No está permitido el uso de teléfonos móviles en el aula durante el período de evaluación continua, excepto indicación expresa en sentido contrario del profesor. Los ordenadores portátiles podrán utilizarse únicamente para actividades relacionadas con la asignatura. El profesor podrá retirar el derecho al uso del ordenador a aquellos alumnos que lo utilicen para actividades que no estén relacionadas con la asignatura (consulta de correos, noticias o redes sociales, consulta o elaboración de actividades de otras asignaturas, etc.).
- No está permitido consumir bebidas ni comidas en el aula. Tampoco está permitida la presencia de cualquier tipo de bebida en las mesas, incluso en envases cerrados.

- Se demandará del alumno una participación activa, necesaria para el desarrollo de las clases.
- Se exigirá al alumno un buen comportamiento en todo momento durante el desarrollo de las clases. El mal comportamiento que impida el normal desarrollo de la clase puede conllevar la expulsión del aula por un tiempo a determinar por el profesor.

7. BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía Básica:

- G. Myers, "The Art of software testing", Wiley John + Sons, ISBN: 978-1118031964
- C. Kaner, J. Falk, H.Q. Nguyen, "Testing Computer Software", Wiley John + Sons, ISBN: 978-0471358466
- D. Graham, R. Black, E. van Veenendaal, "Foundations of software testing", Cengage Learning EMEA, ISBN: 978-1473764798

Bibliografía Recomendada:

- R. Pressman, B. Maxim, "Software Engineering: A Practitioner's Approach", Ninth edition, McGraw-Hill Education, ISBN: 978-1260548006
- K. Beck, "Test Driven Development. By Example", Addison-Wesley, ISBN: 978-0321146533
- K. Beck, "Extreme Programming Explained: Embrace Change" , Addison-Wesley, ISBN: 978-0321278654
- ISO/IEC/IEEE 29119-1 "Software and systems engineering — Software testing"

8. MATERIAL, SOFTWARE Y HERRAMIENTAS NECESARIAS

Editor de texto: Notepad ++
TBD: posiblemente Selenium Web, Cucumber