



GUÍA DOCENTE

VERIFICACIÓN DE SOFTWARE

DOBLE GRADO EN INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2025-2026

Denominación de la asignatura:	Verificación de Software
Titulación:	Doble Grado en Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Optatividad
Curso:	4
Cuatrimestre:	2
Carácter:	OP
Créditos ECTS:	3
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Alonso Alvarez / alonso.alvarez@ext.live.u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

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

Descripción de la asignatura

Esta asignatura es un acercamiento a la calidad del software. Por ello, se centra en entender los conceptos básicos de la calidad y sus 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 ofrecer una primera experiencia aplicable en el desempeño profesional. Esas técnicas y prácticas se basan en estándares de la industria a la hora de diseñar y especificar casos de prueba, así como a su ejecución manual o automatizada.

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE DE LA MATERIA

Competencias (genéricas, específicas y transversales)

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.

Resultados de aprendizaje

Al acabar la titulación, el graduado o graduada será capaz de:

- Entender el ciclo de aseguramiento de la calidad del software
- Diseñar un plan de prueba de software
- Conocer los entornos de prueba más habituales de la industria
- Desarrollar una aplicación intensiva en el uso de GPU
- Ser capaz de medir rendimiento en aplicaciones distribuidas.

CONTENIDO

Prueba y validación de software.

Calidad del software.

Mantenimiento e ingeniería inversa.

TEMARIO

Tema 1. Fundamentos de Calidad del Software

1.1. ¿Qué es Calidad?

1.2. Calidad en sw

1.3. Conceptos básicos: QA, QC, testing

1.4. Verificación del software o testing

1.5. Los siete principios del testing

1.6. El testing como proceso

Tema 2. QA en el SDLC

2.1. Calidad del software en desarrollo predictivo

2.2. Calidad del software en paradigma ágil

2.3. Mantenimiento y soporte

2.4. Test Cases a partir de USsTema

3. Revisión y pruebas

3.1. Tipos de pruebas

3.2. Pruebas de caja blanca

3.3. Análisis estático de código

3.4. Revisión

3.5. Pruebas de caja negra

3.6 Pruebas basadas en experiencia

Tema 4. Técnicas y herramientas

4.1. Herramientas para el testing

4.2. Modelos orientados a pruebas: TDD, BDD, ATDD

4.3. Automatización de pruebas

4.4. CI/CD y DevOps

Tema 5. Gestión de las pruebas

5.1. Los roles del testing

5.2. Reporting y control. Trazabilidad

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
<i>Clases teóricas / Expositivas</i>	16	16
<i>Clases Prácticas</i>	11	11
<i>Tutorías</i>	2	1
<i>Estudio independiente y trabajo autónomo del alumno</i>	25	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	18	0
<i>Actividades de Evaluación</i>	3	3
<i>Preparación y defensa del TFG</i>	<<7- Preparación y defensa del TFG>>	<<Horas presenciales 7- Preparación y defensa del TFG>>

Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo

Método expositivo o lección magistral

Método del caso

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología flipped classroom o aula invertida

Gamificación

DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS PERÍODO TEMPORAL

Tema 1. Fundamentos de Calidad del Software 2-3 semanas

Tema 2. QA en el SDLC 3 semanas

Tema 3. Revisión y pruebas 4-5 semanas

Tema 4. Técnicas y herramientas 3 semanas

Tema 5. Gestión de las pruebas 1 semanas

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	30	60
<i>Prueba Objetiva</i>	30	60
<i>Evaluación del TFG</i>	<<4-(MIN)Evaluación del TFG>>	100

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	60	60

<i>Prueba Objetiva</i>	30	30
<i>Evaluación del TFG</i>	<<4-(MIN)Evaluación del TFG>>	100

Consideraciones generales acerca de la evaluación

- La “Participación activa” se ponderará con un 10% de la nota final. “Participación activa” no significa venir a clase o “ganar” ejercicios colectivos. Se valora la resolución voluntaria de ejercicios y exposiciones. Asimismo, se valorará, como parte de este apartado, la exposición de ideas, la participación en debates, la presentación de propuestas o ejercicios adicionales y, en general, todo lo que demuestre una involucración en la asignatura, y no la mera asistencia pasiva.
- A lo largo del curso se plantearán actividades, ejercicios y trabajos que deberán ser entregados antes de la fecha indicada a través de la plataforma virtual. El conjunto de trabajos supondrá un 60% de la calificación definitiva. Es necesario superar este apartado con un 5 para aprobar la asignatura, y un 4 para liberarlo en convocatoria extraordinaria.
- 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.
- No serán evaluados trabajos entregados fuera de forma y plazo sin causa justificada.
- El examen final valdrá un 30%. Es necesario superar el examen con un 5 para aprobar la asignatura, y un 4 para liberarlo en convocatoria extraordinaria.
- En el caso de que un alumno o alumna no haya conseguido alcanzar el 5 en proyectos y examen, pero tenga al menos un 4 en ambos apartados, se le asignará un trabajo adicional especial a determinar que cubra los aspectos más deficientes de las entregas previas.
- Sólo se podrá superar la asignatura si la nota promedio supera una calificación de 5.0, cumpliendo los requisitos anteriores. En caso de no cumplirse alguno de estos requisitos, la asignatura se considerará automáticamente suspensa independientemente del resto de calificaciones.
- Aquellos alumnos que suspendan los trabajos en la Convocatoria Ordinaria, tendrán la posibilidad de repetirlos en la Extraordinaria.
- Cualquier escrito que el alumno presente (problemas, exámenes, comentarios de los programas, etc.) deberá estar bien presentado, correctamente redactado (con las comas, puntos y puntos y aparte en su lugar adecuado) y sin faltas ortográficas. La nota del escrito podrá bajar hasta un 20% en caso contrario, ya que a un universitario se le exige calidad máxima en su expresión escrita.
- Las notas de los exámenes y del proyecto no se guardan entre cursos académicos sucesivos.
- No es posible obtener Matrícula de Honor (MH) en la Convocatoria Extraordinaria.

- En los exámenes no se permite el uso de apuntes. Teléfonos móviles, ni de calculadoras de ningún tipo, para lo que el alumno debe remitirse a las instrucciones específicas del profesor sobre este tema.
- La asignatura COMPLETA estará suspensa si se descubre que en el examen el alumno copia a otro (ambos estarán suspensos) o bien copia de un libro o de Internet. Además, la universidad abrirá expedientes disciplinarios a ambos alumnos, pudiendo desembocar incluso en su expulsión.

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"

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula

Aula teórica

Equipo de proyección y pizarra

Materiales:

Ordenador personal

Software:

Editor de texto: Notepad ++

Selenium Web, Cucumber

