GUÍA DOCENTE

VERIFICACIÓN DE SOFTWARE

GRADO EN **INGENIERÍA DEL SOFTWARE**

MODALIDAD: A DISTANCIA

CURSO ACADÉMICO: 2023-2024

Denominación de la asignatura:	Verificación de Software
Titulación:	Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Optatividad
Curso:	4º
Cuatrimestre:	2
Carácter:	OP
Créditos ECTS:	3
Modalidad de enseñanza:	A distancia
Idioma:	Castellano
Profesor / Email:	Alonso Álvarez / alonso.alvarez@ext.live.u-tad.com Rafael Socas / rafael.socas@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 enentender los conceptos básicos de la calidad y sus características especiales cuandose aplica al mundo del software.Se da especial relevancia al análisis de casos de uso, la planificación y seguimientode 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 elmundo de la calidad del software para ofrecer una primera experiencia aplicable enel desempeño profesional. Esas técnicas y prácticas se basan en estándares de laindustria a la hora de diseñar y especificar casos de prueba, así como a su ejecución manual o automatizada.

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

Competencias (genéricas, específicas y transversales)

COMPETIENCIAS 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 toleracia 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 Sofware

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 DE APRENDIZAJE

Actividades formativas

Actividad Formativa	Horas totales	Horas síncronas
Sesiones teóricas virtuales síncronas	3,00	3
Sesiones teóricas virtuales asíncronas	11,00	0
Sesiones prácticas virtuales síncronas	2,00	2
Sesiones prácticas virtuales asíncronas	3,00	0
Debate y discusión oral y/o escrita.	5,00	0

Tutorías	2,00	2
Estudio independiente y trabajo autónomo del alumno	25,00	0
Elaboración de trabajos (en grupo o individuales	19,00	0
Actividades de Evaluación	3,00	3
Test de autoevaluación	2,00	0
Prácticas externas	0,00	0
Preparación y defensa virtual del TFG	0,00	0
Seguimiento de proyectos	0,00	0
TOTAL	75	10

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

Presentación - semana 1

Unidad 1 - semana 2-5

Unidad 2 - semana 6-9

Unidad 3 - semana 10-12

Repaso - semana 13-14

Evaluación - semana 15

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	0	30
Evaluación de trabajos, proyectos, informes, memorias	0	70
Prueba Objetiva	0	50

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura	20	10
Evaluación de trabajos, proyectos, informes, memorias	20	20
Prueba Objetiva	60	70

Consideraciones específicas acerca de la evaluación

Será necesario que obtener una nota mínima de 4 puntos (sobre 10) en la prueba final presencial para que se realice la media con las actividades formativas.

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

Materiales:

Ordenador personal

Software:

Editor de texto: Notepad ++

Selenium Web, Cucumber