

## Test y Calidad del Software

Código: 102758  
Créditos ECTS: 6

Titulación	Tipo	Curso	Semestre
2502441 Ingeniería Informática	OB	3	1
2502441 Ingeniería Informática	OT	4	1

### Contacto

Nombre: Xavier Otazu Porter

Correo electrónico: xavier.otazu@uab.cat

### Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)

Algún grupo íntegramente en inglés: No

Algún grupo íntegramente en catalán: Sí

Algún grupo íntegramente en español: No

### Equipo docente

Antonio Ballester Valiente

### Prerequisitos

No es obligatorio, pero se recomienda cursar en paralelo la asignatura "Requisitos del Software" y "Diseño de Software".

### Objetivos y contextualización

En esta asignatura se introducen los conceptos básicos para planificar pruebas del software y las técnicas para implementarlas, así como estimaciones de su calidad.

### Competencias

- Ingeniería Informática
- Adquirir hábitos de trabajo personal.
- 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.
- 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

1. Aplicar las técnicas formales de verificación del cumplimiento de requisitos de las aplicaciones software.
2. Aplicar métodos formales de revisión del diseño del software.

3. Evaluar de forma crítica el trabajo realizado.
4. Evaluar los sistemas software asegurando 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.
5. Planificar y documentar las pruebas del diseño del software en casos prácticos.

## Contenido

El orden temporal de los siguientes contenidos a lo largo del curso no debe seguir necesariamente este orden.

Introducción a la prueba del software

Objetivo de la prueba. Tipo de pruebas: caja blanca, caja negra, de validación, regresión, carga etc. Niveles de prueba: unidad, integración, sistema, aceptación.

Pruebas de unidad y pruebas de integración

Objetivo y metodología. Porque hacer pruebas de unidad e integración? Pruebas de unidad y Test Driven Development. "Mock" objects.

Pruebas de caja negra

Clases de equivalencia. Análisis de valores límite. Tablas de decisión. Pairwise testing. Transición de estados. Diseño de casos de prueba a partir de casos de uso. Ejemplo ATM. Prueba a partir de casos de uso.

Pruebas de caja blanca

Control de flujo (complejidad ciclomática). Cadenas definición-os. Prueba de cobertura para con prueba de control de flujo.

Prueba exploratoria

Pruebas sencillas y rápidas. Interés práctico y efectividad.

Automatización de las pruebas

Creación de pruebas "automáticas". Herramientas scrpting y recording web (Selenium).

Pruebas de sistema o pruebas End-to-End

Objetivo y metodología. Porque hacer pruebas de sistema? Behaviour Driven Development (BDD). Revisión de Test Reports.

Test Frameworks

Definición y ejemplos. Implementación de un test framework y pruebas de sistema utilizando Cucumber (BDD).

Planificación y documentación de la prueba

Análisis del riesgo. Mastertestplan. Ejemplos de plan de pruebas y de documentación.

Quality control, Quality Assurance yQuality Engineering

Revisiones formales, métricas, estándares de desarrollo.

Definición de bug e informes de bugs. Herramientas deBug Tracking: Bugzilla y Jira. Cómo se integra un Quality Engineer en un equipo de desarrollo. Revisiones técnicas formales.

Ecosistema de un Quality Engineer

Definición del ciclo devops. Continuous Integration, Continuous Testing, Continuous Delivery, Continuous Feedback. Herramientas más conocidas (Jenkins, GitHub / GitLab y revisiones de peticiones de cambio, Grafana).

## Metodología

Esta asignatura tiene un equilibrio entre la parte teórica y la parte práctica. No es una asignatura totalmente ABP ya que los conceptos teóricos son importantes, y la parte práctica se realizará con sesiones de tutorías, sesiones de entregas y evaluación de los problemas planteados y trabajo autónomo por parte del alumno.

Clases de teoría: Son clases de tipo magistral donde se introducen los conceptos teóricos de la asignatura. A pesar de ser clases de teoría, una buena parte de la sesión (aproximadamente la mitad) se destinará a solucionar en grupo sencillos problemas planteados por el profesor.

Seminarios: Son clases donde se plantearán casos prácticos que requerirán la aplicación de los conceptos explicados en las clases de teoría. Al final de la clase de seminarios se planteará algún problema que el alumno deberá solucionar de forma autónoma y que entregará voluntariamente al comenzar la siguiente sesión de seminarios.

Laboratorios: Son sesiones en las que el profesor realizará tutorías con los alumnos para guiar su trabajo al desarrollar la solución práctica un problema real de una cierta complejidad. Antes de estas sesiones, el alumno deberá realizar una serie de tareas de forma autónoma que serán mostreadas al comenzar la sesión de laboratorios.

La competencia transversal asignada, T02.08 - Evaluar de manera crítica el trabajo llevado a cabo, se evaluará de manera automática en las prácticas, ya que esta es una capacidad necesaria para poder realizar una práctica de manera autónoma.

Nota: se reservarán 15 minutos de una clase dentro del calendario establecido por el centro o por la titulación para que el alumnado rellene las encuestas de evaluación de la actuación del profesorado y de evaluación de la asignatura o módulo.

## Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Clases de teoría	22	0,88	1
Tipo: Supervisadas			
Laboratorios	12	0,48	1, 2, 3, 4, 5
Seminarios	12	0,48	1, 3, 4, 5
Tipo: Autónomas			
Estudio para pruebas de evaluación	60	2,4	1
Preparación de laboratorios	24	0,96	1, 4, 5
Preparación de seminarios	14	0,56	1, 4

## Evaluación

La nota final de la asignatura viene dada por la fórmula:  $\text{Nota} = 0.45 * \text{Nteo} + 0.55 * \text{NLAB}$ ,

Nteo es la nota de Teoría, y NLAB la nota de laboratorios. Cada una de las notas Nteo y NLAB deben ser superiores a 5, de lo contrario la calificación final de la asignatura será el valor mínimo de estas dos. El valor de Nota sólo se calculará en caso de que Nteo y NLAB sean ambas superiores a 5. En caso de que no sea así, el valor de Nota será el valor mínimo de Nteo y NLAB.

La nota Nteo es

$$\text{Nteo} = 0.5 * \text{Nteo1} + 0.5 * \text{Nteo2}$$

donde Nteo1 y Nteo2 son dos actividades de evaluación que se realizarán a lo largo del curso en horarios de sesiones de teoría. El valor de Nteo sólo se calculará en caso de que Nteo1 y Nteo2 sean ambas superiores a 4. En caso de que no sea así, se considera que el valor de Nteo es el valor mínimo de Nteo1 y Nteo2. Si Nteo no es superior a 5 después de realizar las dos pruebas Nteo1 y Nteo2, el alumno se podrá volver a evaluar de Nteo1 y / o Nteo2 durante las pruebas de evaluación final. En las pruebas de evaluación final, cada una de las notas Nteo1 y Nteo2 se saturarán en un valor de 6

En el mismo momento de publicación de cada evaluación se dará un periodo para revisar la nota. Se respetarán escrupulosamente los períodos de revisión de calificaciones, es decir, no se revisará ninguna calificación fuera de este periodo de revisión de nota.

La nota NLAB será una media ponderada de las diferentes sesiones de laboratorio NLAB (i).

$$\text{NLAB} = w1 * \text{NEntr1} + w2 * \text{NEntr2}$$

donde NEntr (i) es la nota individual de cada una de las dos entregas de prácticas (previsiblemente a las sesiones de laboratorio 3 y 6 del curso, respectivamente), y w1 y w2 son los pesos de cada entrega (se anunciarán los valores el primer día de clase). El cálculo de estas notas está especificado en el enunciado de cada práctica (que se publicará en las aulas Moodle)

No hay segunda convocatoria para la nota NLAB ni por ninguna de las notas NLAB (i). El valor de NLAB sólo se calculará en caso de que todas y cada de NLab (i) sean iguales o superiores a 0,1. En caso contrario NLAB será el valor mínimo de entre todos los valores NLAB (i).

El orden de asignación a los grupos de prácticas se realizará según la nota del expediente académico.

Se convalidará la nota NLAB de aquellos alumnos que hayan cursado la asignatura en un máximo de un curso anterior. No se convalidará otra nota.

Un alumno obtendrá un "No Evaluable" como calificación final de la asignatura únicamente en el caso de que no obtenga ninguna calificación en ninguna de las variables que aparecen en las fórmulas matemáticas anteriores. Si un alumno presenta algún trabajo para evaluar o realiza alguna de las actividades de evaluación que se contemplan en las fórmulas anteriores, ya no podrá obtener un "No Evaluable".

Para obtener una Matrícula de Honor es necesario, como requisito obligatorio, obtener una calificación final de Sobresaliente.

Las fechas de evaluación continua y entrega de trabajos se publicarán en el campus virtual (CV) o aulas Moodle (MO) y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias. Siempre se informará al CV / MO sobre estos cambios ya que se entiende que el CV / MO es el mecanismo habitual de intercambio de información entre profesor y estudiantes.

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, las irregularidades cometidas por un estudiante que puedan conducir a una variación de la calificación se calificarán con un cero (0). Por ejemplo, plagiar, copiar, dejar copiar, ..., una actividad de evaluación, implicará suspender esta actividad de evaluación con un cero (0). Las actividades de evaluación calificadas de esta forma y por este procedimiento no serán recuperables. Si es necesario superar cualquiera

de estas actividades de evaluación peraprovar la asignatura, esta asignatura quedará suspendida directamente con una qualificació de cero (0), sin oportunidad de recuperarla en el mismo curso.

Estas irregularidades incluyen, entre otros:

la copia total o parcial de una práctica, informe, o qualsevolaltra actividad de evaluación;

dejar copiar;

presentar un trabajo de grupo no hecho íntegramente por los miembros del grupo (aplicado a todos los miembros, no sólo a los que no han trabajado);

presentar como propios materiales elaborados por un tercero, aunque sean traducciones o adaptaciones, y en general trabajos con elementos no originales y exclusivos del estudiante;

tener dispositivos de comunicación (como teléfonos móviles, smart watches, bolígrafos con cámara, etc.)

accesibles durante las pruebas de evaluación teórico-prácticas individuales (exámenes);

hablar con compañeros durante las pruebas de evaluación teórico-prácticas individuales (exámenes);

copiar o intentar copiar de otros alumnos durante las pruebas de evaluación teórico-prácticas (exámenes);

usar o intentar usar escritos relacionados con la materia durante la realizaciónde las pruebasde evaluación teórico-prácticas (exámenes), cuando éstos no hayan sido explícitamente permitidos.

A un estudiante que haya cometido irregularidades en un acto de evaluación no se le convalidará

## Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Evaluación laboratorios	0.55	0	0	1, 2, 3, 4, 5
Evaluación continua individual Teoría 1	0.225	3	0,12	1
Evaluación continua individual Teoría 1	0.225	3	0,12	1

## Bibliografía

Entre [ ] los temas que cubren.

[1] A Software Testing Primer: an Introduction to Software Testing. Nick Jenkins.

<http://www.nickjenkins.net>Lectura obligada!

[1,2,3,5,6] Manage Software Testing. Peter Farrell-Vinay. Auerbach Publications, 2008.

Esta bien la parte de documentación y planning, però la de tècnicas de prova no tanto.

[1,3,4] A practitioner's guide to software testing. Lee Copeland, 2004.

Libro muy sencillo y breve (quizás demasiado), enfoque práctico.

[4] How to break software: a practical guideto testing. James A. Whittaker. Addison Wesley, 2003.

EL libro sobre prueba exploratoria.

[5] Pragmatic Unit Testingin Java with Junit. Andy Hunt, Dave Thomas.The Pragmatic Programmer, 2003.

[6] IEEE Standard for Software and System Test Documentation IEEE Std82-2008. IEEE Computer Society, 2008.

[6] Systematic Software Testing by. RickD. Craig, Stefan P.Jaskiel. ArtechHouse, 2002.

El mejor libro que he encontrado sobre la planificación de pruebas. Incluye una explicación del estándar IEEE de documentación.

[7] Llibre EUIS, veure a dalt.

[8] Why the Vasa sank: 10 problems and some antidotes for software projects. Richard E. Fairley, Mary Jane Willshire. IEEE Software, març-abril 2003.

Lectura interesante y divertida.

[9]Video "Scenes of software inspections. Video dramatizations for the classroom." L.Deimel.Software Engineering Institute,CMU/SEI-91-EM-5-0,1991.

[10] Thinking i Java, 3rd edition. B. Eckel. Versión electrònica a <http://www.planetpdf.com/> . Versión impresa en castellano de Pearson Educación, 2002.

Llibre de referència para programación en Java (pràctica).

[5] <https://testautomationu.applitools.com/junit5-tutorial/> Tutorial sobre JUnit.

[5] <https://testautomationu.applitools.com/unit-testing/index.html> Introducció al Unit Testing.

[5] <https://site.mockito.org/> Paqueta para implementar mock object.

[7] <https://testautomationu.applitools.com/codeless-test-automation-with-selenium-ide/> Tutorial sobre selenium

[7] <https://testautomationu.applitools.com/cucumber-java-tutorial/> Tutorial sobre cucumber.

## **Software**

JUnit (<https://junit.org/junit5/>)

Bugzilla (<https://www.bugzilla.org/>)

Jira (<https://www.atlassian.com/software/jira>)

Selenium (<https://www.selenium.dev/>)

Cucumber (<https://cucumber.io/>)

GitHub/GitLab (<https://github.com/>) (<https://about.gitlab.com/>)