

## Laboratorio Integrado de Software

Código: 102788  
Créditos ECTS: 9

Titulación	Tipo	Curso	Semestre
2502441 Ingeniería Informática	OB	3	2
2502441 Ingeniería Informática	OT	4	2

La metodología docente y la evaluación propuestas en la guía pueden experimentar alguna modificación en función de las restricciones a la presencialidad que impongan las autoridades sanitarias.

### Contacto

Nombre: Antonio Manuel López Peña

Correo electrónico: AntonioManuel.Lopez@uab.cat

### Uso de idiomas

Lengua vehicular mayoritaria: español (spa)

Algún grupo íntegramente en inglés: No

Algún grupo íntegramente en catalán: No

Algún grupo íntegramente en español: No

### Equipo docente

Jose Luis Gomez Zurita

### Equipo docente externo a la UAB

Pablo Torrellas Pérez

### Prerequisitos

Esta asignatura tiene un carácter muy práctico, pero se basa en los fundamentos teóricos impartidos en otras asignaturas de la mención en Ingeniería del Software. Por lo tanto, si se quiere cursar esta asignatura con éxito, es necesario haber cursado previamente:

- (1) Requisitos del software,
- (2) Diseño de software,
- (3) Gestión y Administración de Bases de Datos,
- (4) Test y Calidad del Software,
- (5) Gestión del Desarrollo de Software (esta se puede cursar paralelamente, no es necesario haberla cursado previamente).

### Objetivos y contextualización

Esta asignatura quiere proporcionar una experiencia útil al alumnado respecto a lo que se pueden encontrar a la hora de desarrollar proyectos software profesionalmente. Por lo tanto, desde el punto de vista del alumnado, se trata de desarrollar un proyecto software completo y relativamente largo. Además, el proceso de desarrollo será tan o más importante que el resultado final (producto software). Por tanto, el software se deberá desarrollar de la forma más profesional posible, en particular, aplicando las mejores prácticas y trabajando en equipo.

Así, los objetivos de la asignatura son:

1. Trabajar en un equipo relativamente grande, compuesto de sub-equipos con diferentes responsabilidades.
2. Aplicar los conocimientos teóricos que constituyen las mejores prácticas del desarrollo de software.
3. Desarrollar un software completo, es decir, desde el papel en blanco hasta una aplicación, con su documentación interna y externa, y que satisfaga los requisitos de un cliente.
4. Que el desarrollo de este software también sirva para adquirir experiencia en entornos SW actuales.

La aplicación desarrollada, y la manera en que se ha desarrollado, debería poder servir como "carta de presentación" curricular.

## Competencias

### Ingeniería Informática

- Adquirir hábitos de pensamiento.
- Adquirir hábitos de trabajo personal.
- Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.
- Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse.
- 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.
- 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.
- Trabajar en equipo.

## Resultados de aprendizaje

1. Adaptarse a situaciones imprevistas.
2. Asumir y respetar el rol de los diversos miembros del equipo, así como los distintos niveles de dependencia del mismo.
3. Definir y gestionar la documentación que se genera durante el desarrollo de una aplicación software.
4. Desarrollar el pensamiento sistémico.
5. Diseñar una arquitectura de un sistema informático basado en componentes.
6. Diseñar una arquitectura que permita solucionar de manera óptima el problema especificado, teniendo en cuenta los riesgos asociados.
7. Evaluar de forma crítica el trabajo realizado.
8. Gestionar el tiempo y los recursos disponibles. Trabajar de forma organizada.
9. Identificar, gestionar y resolver conflictos.
10. Planificar la integración de las diferentes componentes desarrolladas en el proceso de codificación.
11. Resolver problemas de integración del software con iniciativa y autonomía.
12. Saber comunicar y transmitir conocimientos, habilidades y destrezas relativos a la integración del software.
13. Seleccionar y utilizar las herramientas CASE adecuadas a cada fase de desarrollo del software.
14. Tomar decisiones propias.
15. Trabajar cooperativamente.

## Contenido

En esta asignatura no buscamos añadir más contenido teórico a lo que ya se desarrolla en las otras asignaturas de la mención de Ingeniería del Software. Es decir, ya se asumen los conocimientos teóricos de la captura de requisitos, el diseño y codificación orientado a objeto, la gestión y administración de bases de datos, el test y calidad del software, y la gestión del desarrollo. Se trata de poner todo esto en práctica para desarrollar una aplicación concreta. Por lo tanto, como contenidos nuevos hay dos aspectos:

1. El trabajo en equipo: roles y responsabilidades de un equipo software, reuniones, presentaciones, etc.
2. Métodos ágiles de desarrollo de software.

## Metodología

Esta asignatura se enfoca a desarrollar una aplicación software de la forma más realista posible. Por lo tanto, se hace un aprendizaje basado en un caso práctico. De hecho habrá un caso práctico para cada equipo de prácticas. Un equipo estará formado por un conjunto de entre 6 y 9 estudiantes. Por un lado esto significa que habrá mucho trabajo autónomo, pero, por otro lado, también habrá mucho trabajo de equipo. Así, la tipología de las clases apoyará el trabajo de equipo y el desarrollo de software basado en un proceso ágil. En particular, tendremos:

Clases de teoría. Están destinadas a complementar los aspectos no cubiertos por las otras asignaturas de la mención de Ingeniería del Software, principalmente el trabajo en equipo.

Reuniones de proyecto. Cualquier equipo que desarrolle una aplicación determinada, estará dividido en sub-equipo con diferentes responsabilidades (p.e., jefes de proyecto, responsables de pruebas, responsables de diseño y codificación, etc.). Estos sub-equipo deben reunirse y discutir el estado del proyecto desde los diferentes puntos de vista.

Clases de presentación. Cada equipo tiene que hacer presentaciones públicas (para el resto de equipos) regulares sobre el estado de su proyecto.

Todas las comunicaciones con el alumnado se harán mediante los mecanismos habilitados en el campus virtual.

## Actividades

Título	Horas	ECTS	Resultados de aprendizaje
<b>Tipo: Dirigidas</b>			
Clases de teoría	15	0,6	2, 15
Presentaciones	16	0,64	7, 12
Reuniones de proyecto	40	1,6	1, 2, 7, 3, 6, 9, 10, 11, 13, 15
<b>Tipo: Autónomas</b>			
Estudio para los exámenes individuales	16	0,64	7
Tareas asignadas al proyecto	134	5,36	1, 2, 7, 3, 4, 5, 6, 8, 9, 10, 14, 11, 13, 15

## Evaluación

La evaluación se basará en las notas siguientes:

1) Desarrollo de la aplicación: cabe destacar que no sólo se valora el resultado final, es decir, la aplicación en sí misma, sino que también se valora, y mucho, el proceso de desarrollo de la misma. Por lo tanto, tendrá mucha importancia la evaluación continua del trabajo hecho en las reuniones de proyecto y, especialmente, en el estado del desarrollo en hitos preestablecidos. En general, la mayor parte de esta nota será común a todo el equipo, pero se establecerán mecanismos para garantizar que todos los miembros del equipo realmente merecen la nota. Esto se hará a partir de evaluaciones entre compañeros/as, que también servirán para terminar de configurar el total de la nota DA. Si se detecta algún caso en que un/a estudiante no haya

hecho su trabajo o tiene una contribución marginal, el/la estudiante involucrado/a puede recibir una nota inferior, incluso, dado que estamos ante una evaluación continua, el/la estudiante puede suspender directamente la asignatura (nota de cero). En particular hay dos pruebas parciales de este tipo, DA1 y DA2.

2) Presentaciones de equipo: además del material requerido para hacer la evaluación tipo DA, habrán presentaciones regulares del trabajo (semanales o quincenales) para toda la clase. Estas presentaciones serán evaluadas como tal, y pueden ser entendidas como una prueba de equipo, por lo tanto, la nota será para todo el equipo. Respecto a la implicación de los miembros del equipo, se aplica el mismo criterio que en el caso DA. En particular hay dos pruebas parciales de este tipo, PE1 y PE2.

3) Exámenes individuales: se harán exámenes individuales para evaluar los conocimientos teórico-prácticos que cada estudiante. En particular hay dos pruebas parciales de este tipo, EI1 y EI2.

Para aprobar la asignatura, deben aprobarse por separado todos estos apartados. Es decir,  $DA > = 5$  y  $PE > = 5$  y  $EI1 > = 5$  y  $EI2 > = 5$ , donde DA, PE, EI1 y EI2 son notas sobre 10. Si esto se cumple, entonces la nota final, NF, se calcula como:

[Oportunidad 1]  $NF = 0.3 DA1 + 0.5 DA2 + 0.05 PE1 + 0.05 PE2 + 0.05 EI1 + 0.05 EI2 + Extra.$

ATENCIÓN: Si  $EI1 < 5$  o  $EI2 < 5$ , el/la estudiante puede tener una segunda oportunidad de aprobar la asignatura mediante un examen individual de recuperación, pero solo si se cumplen ciertas condiciones. En particular, si  $DA = (5/4) * (0.3 DA1 + 0.5 DA2)$  y  $PE = 0.5 * (PE1 + PE2)$ , para poder hacer este examen se exige que  $DA > = 5$  y  $PE > = 5$ . Entonces, si ER es la nota del examen de repesca, la nota final de la asignatura la calcularemos como:

[Oportunidad 2, para recuperar Exámenes Individuales]  $NF = \min (6, 0.5 DA + 0.5 ER).$

ATENCIÓN: Si  $DA < 5$  o  $PE < 5$  el/la estudiante recibirá una nota final (NF) de entre 0 y 4.5 (suspenso) dependiendo del caso. Este/a estudiante no tendría "segunda oportunidad". Nótese, que la nota máxima que puede recibir un/a estudiante en "segunda oportunidad" es un 6.

Se entiende que los/las estudiantes "No evaluables" sólo son aquellos/as que no se han sometido a ninguna actividad de evaluación.

El número de MH (Matrícula de Honor) que se puede conceder es proporcional al número de estudiantes matriculados/as. Para poder obtener MH el/la estudiante debe cumplir  $NF > 9.5$  al momento de la Oportunidad 1. Digamos que se pueden otorgar "N" MH. A partir de ahí, si hay más de N candidatos/as, el profesor examinará la trayectoria de los/las estudiantes candidatos/as a MH y seleccionará los/las N. Por lo tanto, es el profesor quien evalúa las evidencias que considere más oportunas (trabajo continuado, relevancia dentro del equipo, NF, etc.) para tomar la decisión final.

Las aplicaciones que se desarrollan son propuestas por los/las estudiantes de forma individual y salen de un proceso de selección basado en presentaciones y votaciones. Por lo tanto, durante esta primera fase, las competencias transversales T01 y T02 son muy importantes. De hecho, el/la estudiante que propone una aplicación que pasa todo el proceso (es decir, se convierte en una aplicación que se llevará a cabo por un equipo), es nombrado jefe/a de equipo y recibe 1 punto "Extra" (ver NF en Oportunidad 1). Hablando de competencias transversales, nótese que toda la asignatura está inherentemente ligada a T03. Pero, DA y PE cuantifican esta competencia en particular. Además, el/la jefe/a de equipo puede distribuir 1 punto "Extra" (ver NF en Oportunidad 1) entre los miembros del equipo con una contribución más significativa vista desde dentro del equipo.

Cabe destacar también que los/las repetidores/as no recibirán ningún trato especial, deben cursar la asignatura como el resto de estudiantes.

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, y de acuerdo con la normativa académica vigente, se calificarán con un cero las irregularidades cometidas por un/una estudiante que puedan conducir a una variación de la cualificación de un acto de evaluación. Por lo tanto, plagiar, copiar o dejar copiar una práctica o cualquier otra actividad de evaluación implicará suspender con un cero y no se podrá recuperar en el mismo curso académico. Si esta actividad tiene una nota mínima asociada, entonces la asignatura quedará suspendida.

## Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Desarrollo de la aplicación (DA1 y DA2)	80% (DA1: 30%, DA2: 50%)	0	0	1, 2, 7, 3, 4, 5, 6, 8, 9, 10, 14, 11, 12, 13, 15
Exámenes individuales (EI1 y EI2)	10% (EI1: 5%, EI2: 5%)	4	0,16	3, 5, 6, 10, 11, 12, 13
Presentaciones de equipo (PE1 y PE2)	10% (PE1: 5%, PE2: 5%)	0	0	12

## Bibliografía

A la bibliografía propia del resto de las asignaturas de la mención de Ingeniería del Software, se debe añadir:

- [1] Practical tips for Software-intensive Student Projects, D.C. Rajapakse.
- [2] Agile Software Development: Principles, Patterns and Practices, R.C. Martin.
- [3] Agile Software Development: The Cooperative Game, A. Cockburn.
- [4] Crystal Clear: A Human-Powered Methodology for Small Teams, A. Cockburn.
- [5] Agile Software Development with Scrum, K. Schwaber, M. Beedle.
- [6] Agile Project Management with Scrum, K. Schwaber.