

Titulación	Tipo	Curso
Ingeniería de Datos	OB	3

## Contacto

Nombre: Juan Francisco Protasio Ramírez

Correo electrónico: joan.protasio@uab.cat

## Equipo docente

Marc Castello Torrellas

Juan Francisco Protasio Ramírez

## Idiomas de los grupos

Puede consultar esta información al [final](#) del documento.

## Prerrequisitos

Dado que la asignatura es de introducción al bloque de asignaturas de ingeniería del software, se asumirá que no se posee ningún tipo de conocimiento previo sobre la materia. Es responsabilidad de la propia asignatura proporcionar a los / las alumnos un medio para adquirir los conocimientos descritos en el apartado de contenidos de la asignatura (apartado 6 de esta guía). A pesar de ello es recomendable haber superado las asignaturas de Fundamentos de Informática y Metodología de la Programación de primer curso, y la asignatura de Bases de Datos de segundo curso.

## Objetivos y contextualización

Esta asignatura es introductoria del bloque de asignaturas sobre el proceso de ingeniería del software. El objetivo general es dar una visión global y ordenada del proceso de desarrollo del software que no sea simplemente el de la programación (que constituye, sólo, una fase dentro de todo el proceso de la ingeniería del software). Se introducirán las actividades fundamentales que constituyen el proceso de desarrollo (análisis de requisitos, diseño, implementación, pruebas y mantenimiento), la mayoría de las cuales se tratan en más profundidad en asignaturas posteriores.

Se trata, por tanto, que el alumno adquiera una visión general de qué es un proceso de desarrollo de software, como se modela software, qué herramientas se utilizan, cómo se gestiona la calidad y cómo se administra un proyecto. Más concretamente, los objetivos son:

Proporcionar una visión general del proceso de desarrollo del software, los paradigmas, las actividades, y los objetivos de cada actividad.

Conocer qué es el modelado del software. La importancia y la dificultad de la especificación de requerimientos y el diseño en sus diferentes vistas.

Aprender los conceptos básicos de UML para trabajar de manera práctica en el modelado del software.

Introducir al alumno / a en los principios, conceptos y técnicas que se aplican para administrar y controlar la calidad del software.

Dar a conocer las técnicas de administración requeridas para planificar, organizar, monitorizar y controlar proyectos de software.

Familiarizarse con el modelo SCRUM como paradigma de desarrollo evolutivo más relevante.

## **Competencias**

- Diseñar soluciones algorítmicas eficientes para problemas computacionales, implementarlas en forma de desarrollo de software robustos, estructurados y fáciles de mantener, y verificar su validez.
- Evaluar de manera crítica el trabajo realizado.
- Planificar y gestionar el tiempo y los recursos disponibles.
- 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.

## **Resultados de aprendizaje**

1. Analizar, diseñar, distribuir y mantener aplicaciones, asegurando su calidad y su mantenibilidad.
2. Analizar el rendimiento de aplicaciones, detectar cuellos de botella y aplicar posibles optimizaciones.
3. Describir un sistema software y transformarlo en un modelo de diseño.
4. Evaluar de manera crítica el trabajo realizado.
5. Planificar y gestionar el tiempo y los recursos disponibles.
6. 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.

## **Contenido**

### **BLOQUE 1. EL SOFTWARE Y LA INGENIERÍA DEL SOFTWARE**

#### **Tema 1. Principios de la ingeniería del software.**

Definición y objetivos de la SE. Definición de software. Características del software. Aplicaciones del software.

Definición de ES. Objetivos del SE.

Evolución del software. Etapas. Crisis del software: problemas y causas.

Proceso, método y herramienta. Definiciones. Actividades en el proceso de desarrollo del software.

Paradigmas del desarrollo del software. Modelo lineal secuencial (ciclo de vida clásico). Modelo de prototipado. Modelo evolutivo. Modelo en espiral.

### **BLOQUE 2. ADMINISTRACIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE**

## Tema 2. SCRUM.

Introducción. Metodología Ágil de gestión de proyectos.  
SCRUM. Características. Historia.  
Componentes de SCRUM. Roles. Artefactos. Procesos.  
Escalabilidad. Scrum de Scrum.  
Herramientas Software.  
Otros modelos ágiles. Lean. Xtreme programming. Kanban.  
Un caso de estudio. Presentación de caso de estudio a seguir durante el curso.

[NOTA: Dado que es necesario la adquisición de los conocimientos de SCRUM de cara a las prácticas, este tema se trata en este momento del curso a pesar de que pertenece al bloque 4 sobre administración de proyectos]

## BLOQUE 3. MODELADO DE SOFTWARE

### Tema 3. Análisis de Requerimientos del Software.

Introducción. Tipo de requerimientos. Tareas a realizar.  
Comprensión del problema. Técnicas de comunicación. Problemas asociados. Principios del análisis.  
Especificación de requerimientos. Propiedades deseables de una ER. Estándares de ES. Revisión y validación de la especificación.

### Tema 4. Diseño del Software.

Introducción. Proceso de diseño. Diseño de datos, diseño arquitectónico, diseño de la interfaz, diseño procedimental. Principios (objetivos) del diseño.  
Conceptos del diseño. Abstracción. Modularidad. Refinamiento.  
Diseño modular efectivo. Independencia funcional. Cohesión. Acoplamiento. Heurísticas para un diseño modular efectivo.  
Diseño de Interfaces de Usuario.

### Tema 5. UML (Unified Modeling Language)

Evolución e historia de UML.  
Vistas de UML. Vista de casos de uso, vista lógica, vista de componentes, vista de desarrollo.

## BLOQUE 4. ADMINISTRACIÓN DE LA CALIDAD DEL SOFTWARE

### Tema 6. Calidad del software.

Introducción. Conceptos de calidad.  
Prueba del software. Estrategias de prueba, prueba de aplicaciones convencionales, prueba de aplicaciones web.  
Administración de la configuración del software.

## BLOQUE 5. ADMINISTRACIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE

### Tema 7. Administración de proyectos del software.

Conceptos de administración de proyectos.  
Planificación.  
Administración del riesgo.  
Mantenimiento y reingeniería.

## Actividades formativas y Metodología

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Clases de teoría	30	1,2	3, 6
Preparación y estudio	30	1,2	4, 5, 6
Seminarios/problemas	15	0,6	1, 2, 4, 3
Tipo: Supervisadas			
Entrevistas y tutorización del caso práctico	15	0,6	1, 2, 4, 3, 5
Tutorías	15	0,6	4, 5
Tipo: Autónomas			
Trabajo práctico	38	1,52	1, 2, 4, 3, 5, 6

La asignatura de Ingeniería del Software tiene una fuerte componente aplicada, pero también requiere de una parte teórica, fruto de la experiencia, a partir de la cual se implementan las decisiones prácticas. Por tanto, la asignatura no será puramente ABP (Aprendizaje Basado en Problemas) sino una mezcla de ABP y método tradicional. Se propondrá al alumno un caso práctico (problema genérico) de dimensión real, alrededor del cual se estructurará las sesiones de problemas y el trabajo práctico. El alumno deberá adquirir el perfil de analista / arquitecto del software para tomar las decisiones oportunas que concluyan en la construcción de una propuesta de diseño del sistema propuesto. El profesor hará el papel de tutor y, en situaciones determinadas, de cliente, fomentando simulaciones de diálogo desarrollador - cliente.

La asignatura consta de 4 horas semanales presenciales. No se distingue entre horarios de teoría, problemas y prácticas de laboratorio. Durante las horas presenciales se alternarán clases de teoría, problemas o prácticas según la planificación que se indicará el día de la planificación de la asignatura. De manera general, se concentrará la explicación de los contenidos teóricos fundamentales durante las primeras sesiones del curso (aproximadamente las 6 primeras semanas), pero también se intercalarán de forma gradual algunas sesiones de problemas y casos prácticos. Una vez impartidos los contenidos teóricos, se dedicarán todas las sesiones restantes a trabajo basado en problemas, identificando algunas sesiones de asistencia obligatoria donde se llevará a cabo la práctica en equipos de 5 personas o se harán problemas a entregar al final de la sesión.

## TEORÍA

Consiste en clases magistrales con material multimedia disponible en el Campus Virtual de la UAB. El objetivo principal de estas clases es introducir las nociones básicas sobre los procesos, métodos y herramientas relativos a la visión del ciclo de vida del software desde un punto de vista de ingeniería. Se expondrá cuáles

son los principios y paradigmas de la ingeniería del software, las diferentes actividades y las tareas a llevar a cabo en cada actividad. Las clases de teoría deben permitir al alumno / a coger una visión real del rigor, planificación y la sistemática que requiere el desarrollo profesional del software. No se profundizará en los temas concretos ya que en asignaturas de cursos posteriores estos ya se desarrollan, pero se explicarán los conceptos necesarios para que el alumno / a entienda el proceso de desarrollo de software completo.

## SEMINARIOS / PROBLEMAS

Los seminarios son clases con trabajo participativo del alumnado. Según el objetivo de los seminarios, se dividirá el alumnado del grupo en subgrupos más reducidos. Tienen una doble función, enlazando por un lado con las clases magistrales y por la otra con las sesiones prácticas sobre el problema a resolver en equipos. Por una parte, los seminarios complementarán los contenidos tecnológicos expuestos en las clases magistrales, completando su comprensión. Por otro lado serán el marco de trabajo donde debatir, establecer las bases y desarrollar el problema guía. Se le darán al estudiante los conocimientos necesarios para llevar adelante el trabajo práctico o se le indicará dónde conseguirlos. En los seminarios se promueve fundamentalmente la capacidad de análisis y síntesis, así como el razonamiento crítico y la toma de decisiones del alumno frente a la resolución del problema guía. Dado que se trata de una asignatura muy práctica y donde los contenidos teóricos son de baja complejidad, se refuerza a menudo el trabajo práctico con resolución de problemas o el trabajo de actividades participativas. Se coordinarán los contenidos de los seminarios con las etapas a resolver en el trabajo de la práctica.

## LOGRO DE COMPETENCIAS

T03 - Trabajar en equipo. Se trabajará en el trabajo práctico donde los alumnos / as formarán equipos de 5-6 personas, y donde se considera el trabajo en equipo como una de las competencias a trabajar. Los alumnos / as deberán distribuirse los roles que habría en un proyecto de software, y cada semana uno de ellos hará las funciones de coordinador (scrum master). Se evaluará como uno de los indicadores del trabajo práctico.

CRI03.03 - Comprender la importancia de la negociación, los hábitos de trabajo en equipo, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo software, a un nivel avanzado: Se trabajará a sesiones específicas de problemas donde se propondrán técnicas de captura de requerimientos, en particular de Gamestorming, y también en la práctica, donde los grupos de trabajo se organizan siguiendo los principios de la metodología SCRUM, y en cada semana (sesión) se requerirá que uno de los miembros del equipo asuma el papel de SCRUM master (coordinador). Se evaluará a partir del seguimiento individualizado del trabajo práctico (a partir de las actas que el Scrum master haga de las reuniones de seguimiento).

CRI08.01 - Diseñar y mantener aplicaciones siguiendo criterios de robustez y fiabilidad y CRI08.02 - Diseñar aplicaciones sabiendo seleccionar el paradigma de desarrollo software más adecuado: Se trabajarán tanto en las sesiones presenciales, en particular de problemas, como el trabajo práctico. Se evaluará con la prueba escrita (examen de teoría, primer control) como con la calificación del trabajo práctico.

CRI16.01 - Comprender qué es el análisis de requerimientos y porque es importante y CRI16.02 - Conocer el proceso de desarrollo del software ágil: Se trabajarán en las primeras sesiones del curso de teoría y problemas (primeras cuatro semanas), así como en las primeras sesiones de prácticas (primeras dos sesiones). Se evaluarán con la prueba escrita (examen de teoría, segundo control) como con la calificación del trabajo práctico.

CRI17.02 - Conocer la notación y construcción de diagramas UML y E03.02 - Diseñar y evaluar interfases software persona-computador que garanticen la accesibilidad y usabilidad de los sistemas, servicios y aplicaciones informáticas: Se trabajarán en las sesiones centrales del curso de teoría y problemas (semana 5 a semana 10), así como a las sesiones de prácticas (sesiones 3 a 5). Se evaluarán con la prueba escrita (examen de teoría, segundo control) como con la calificación del trabajo práctico. Se propondrá a los alumnos ejercicios opcionales que podrán entregar por tener puntuaciones adicionales en el examen correspondiente a esta parte.

E03.01 - Comprender qué es lo que se necesita para conseguir un buen diseño que incluya los requerimientos de accesibilidad, ergonomía, usabilidad y seguridad de las aplicaciones informáticas y E03.02 - Diseñar y evaluar interfases software persona-computador que garanticen la accesibilidad y usabilidad los sistemas, servicios y aplicaciones informáticas: al igual que la competencia anterior, se trabajarán en las sesiones centrales del curso de teoría y problemas (semana 5 a semana 10), así como las sesiones de prácticas (sesiones 3 a 5). Se evaluarán con la prueba escrita (examen de teoría, segundo control) como con la calificación del trabajo práctico.

E05.01 - Entender qué es el software, cuáles son los objetivos que se plantean en desarrollar un software y la problemática asociada a su desarrollo. Se trabajará a las sesiones finales de teoría y problemas (semanas 11

y 12), así como las sesiones de prácticas (sesión 6). Se evaluará con la prueba escrita (examen de teoría, segundo control) como con la calificación del trabajo práctico.

E09.04 - Conocer y aplicar métodos de comunicación y negociación eficaz en la realización de las tareas propias de la profesión. Fundamentalmente se trabajará en las sesiones de prácticas donde los alumnos deberán defender ante el profesor las decisiones tomadas en su proyecto. El / la profesor / a hará el papel de cliente para simular lo que sería un proyecto real. Se evaluará en la calificación del trabajo práctico.

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.

## Evaluación

### Actividades de evaluación continuada

Título	Peso	Horas	ECTS	Resultados de aprendizaje
Evaluación colectiva	60	4	0,16	1, 2, 4, 3, 5, 6
Evaluación individual	40	3	0,12	3, 6

### INDICADORES Y VALORACIÓN:

Esta asignatura no prevee el sistema de evaluación única.

La evaluación se llevará a cabo en base al trabajo práctico desarrollado por el alumno e incluido en su portfolio (resolución final de problema guía, material generado en las actividades de los seminarios), el grado de implicación en los seminarios y los conocimientos tecnológicos alcanzados de la materia a estudio. Se contemplarán los siguientes instrumentos de evaluación:

Evaluación individual (AI). Prueba escrita presencial para valorar el grado de conocimientos alcanzados por el alumno a nivel individual. Habrá dos pruebas parciales durante el curso que permitirán liberar materia siempre y cuando la nota obtenida sea igual o superior a 5. La nota correspondiente a cada parcial tiene un peso del 50% en la nota final de la evaluación individual.

Por lo tanto,  $AI = 0.5 \text{ Nota\_Parcial1} + 0.5 \text{ Nota\_Parcial2}$ .

Evaluación colectiva de los equipos de trabajo (AC). Prueba del rendimiento conjunto de los diferentes grupos de trabajo para resolver el problema guía y las diferentes actividades presentadas en los seminarios. Consta de dos instrumentos:

AC1 (evaluación técnica). Documentación entregada por el alumnado de su trabajo práctico. Se evaluará la resolución técnica del problema guía.

AC2 (evaluación de progreso y competencias). La defensa del trabajo por parte de los alumnos en las entregas, el seguimiento de la evolución del trabajo y la participación activa en los seminarios. Se evaluará a partir de las entrevistas con el profesor, pruebas escritas de corta duración, entregas o exposiciones de problemas y casos planteados en las actividades. Aunque este instrumento es de evaluación colectiva, a criterio del tutor / a, puede ser corregido individualmente en casos en que la participación del alumno / a destaque (por exceso o defecto) respecto a su grupo.

Cabe señalar que mientras los instrumentos AI y AC1 valoran el rendimiento del alumno / a y el grado de consecución de los conocimientos, el instrumento AC2 valora el esfuerzo del alumno / a. Para calcular AC, los pesos de los valores de AC1 y AC2 se indicarán en clase a principio de curso. Para calcular el valor de AC, es necesario que tanto AC1 como AC2 sean superiores a 5 .

La calificación final se obtendrá según el siguiente baremo:

$$QF = 0.4 AI + 0.6 AC$$

Para aprobar la asignatura es necesario haber conseguido una puntuación mínima de 5 en las dos calificaciones (AI y AC), así como en las pruebas parciales para liberar materia establecidos a lo largo del curso. A criterio del profesor / a podrá, sin embargo, establecer compensaciones entre las notas de las entregas parciales de las actividades (correspondientes a AC).

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

La asignatura será evaluada como No Evaluable sólo en el caso que el alumno no se haya presentado a ninguna de las pruebas de evaluación contempladas ni haya entregado total o parcialmente los trabajos.

En caso de no superar la asignatura debido a que alguna de las actividades de evaluación no alcanza la nota mínima requerida, la nota numérica del expediente será el valor menor entre 4.5 y la media ponderada de las notas. Con las excepciones de que se otorgará la calificación de "no evaluable" a los / as estudiantes que no participen en ninguna de las actividades de evaluación, y de que la nota numérica del expediente será cero (0) en caso de que el estudiante haya cometido irregularidades en un acto de evaluación (y por tanto no será posible el aprobado por compensación).

Se otorgarán las Matrículas de Honor dentro de los máximos admitidos por la normativa de la UAB (en función del número de matriculados) a las notas más altas iguales o superiores a 9.

Para cada actividad de evaluación, se indicará un lugar, fecha y horade revisión en la que el estudiante podrá revisar la actividad con el profesor. En este contexto, se podrán hacer reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Si el estudiante no se presenta en esta revisión, no se revisará posteriormente esta actividad.

Ver apartado "PLAGIO" sobremesures en casos de irregularidades por plagio en las actividades de evaluación.

#### RECUPERACIONES:

Evaluación individual (AI). Se harán dos exámenes parciales de teoría liberatorios durante horas lectivas. Los alumnos que no superen esta prueba (con nota igual o superior a 5), dispondrán de un examen de recuperación en la fecha de evaluación final programada para la titulación. En este examen de recuperación, la nota máxima que se puede obtener es de 6.

Evaluación colectiva (AC). El trabajo práctico se evalúa en forma de evaluación continua en las sesiones de seguimiento. Por lo tanto no habrá ninguna actividad de recuperación al final del curso. Sin embargo y con respecto a la resolución del problema guía, el hecho de no entregar a tiempo o no superar una de las entregas intermedias puede ser recuperado reprogramando la planificación del trabajo de las sesiones posteriores durante el curso, de acuerdo con el profesor.

#### FECHAS DE EVALUACIÓN:

Las fechas de evaluación continua y entrega de trabajos se publicarán alcampus virtual y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias. Siempre informará el campus virtual sobre estos cambios ya que se entiende que es el mecanismo habitual de intercambio de información entre profesor y estudiantes.

#### ALUMNOS REPETIDORES:

No se guardan partes parciales (teoría o prácticas) de un curso para otro. Sin embargo, a criterio del profesor y en función de las evaluaciones de cursos previos, se podrán establecer compensaciones.

#### PLAGIO:

Sin perjuicio de otras medidas disciplinarias que 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 para probar la asignatura, esta asignatura quedará suspendida directamente con una calificación 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 cualquier otra actividad de evaluación  
dejar copiar  
presentar un trabajo de grupo no hecho íntegramente por los miembros del grupo  
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  
uso no autorizado de la IA (p. ej., Copilot, ChatGPT o equivalentes) para resolver ejercicios, prácticas y/o cualquier otra actividad evaluable;  
tener dispositivos de comunicación (como teléfonos móviles, smart watches, etc.) accesibles durante los proves de evaluación teórico-prácticas individuales (exámenes).

A un estudiante que haya cometido irregularidades en un acto de evaluación no se le convalidará ninguna de las Actividades de evaluación realizadas. En resumen: copiar, dejar copiar o plagiar (o el intento de) en cualquiera de las actividades de evaluación equivale a un SUSPENSO, con un cero (0) no compensable y sin convalidaciones de partes de la asignatura en cursos posteriores.

#### ACLARACIÓN FINAL:

Para cualquier duda o discrepancia, prevalecerá la información más actualizada que se comunicará el día de la presentación de la asignatura y que se publicará en el campus virtual.

### Bibliografía

- G. Booch, J. Rumbaugh, I. Jacobson. El lenguaje unificado de modelado. Addison-Wesley, 2001.
- I. Jacobson, G. Booch, J. Rumbaugh. El proceso unificado de desarrollo de software. Addison-Wesley, 2000.
- J. Rumbaugh, I. Jacobson, G. Booch. El lenguaje unificado de modelado: manual de referencia. Addison-Wesley, 2007.
- T. Quatrani. Visual Modeling with Rational Rose 2000 and UML. Addison-Wesley, 2000.
- P. Krutchen. The Rational Unified Process. An Introduction. Addison-Wesley, 2000.
- Roger S. Pressman, Ingeniería del software, un enfoque práctico. McGraw-Hill, 7a. edición, 2010.
- A. Álvarez García, R. de las Heras del Dedo, C. Lasa Gómez, Métodos Ágiles y Scrum. Anaya Multimedia, 2012.
- K. Pohl, C. Rupp. Requirements Engineering Fundamentals. Rocky Nook Inc. 2011.

Atès que avui en dia molts materials estan en línia, molt més actualitzats que les fonts bibliogràfiques, durant el curs es proporcionen enllaços a documentació lliure a la xarxa, video-lectures, etc.



## Software

ArgoUML (<http://argouml.tigris.org/>)

Eclipse (<https://www.eclipse.org/ide/>)

## Grupos e idiomas de la asignatura

La información proporcionada es provisional hasta el 30 de noviembre de 2025. A partir de esta fecha, podrá consultar el idioma de cada grupo a través de este [enlace](#). Para acceder a la información, será necesario introducir el CÓDIGO de la asignatura

Nombre	Grupo	Idioma	Semestre	Turno
(PAUL) Prácticas de aula	81	Catalán/Español	primer cuatrimestre	tarde
(PAUL) Prácticas de aula	82	Catalán/Español	primer cuatrimestre	tarde