

Titulación	Tipo	Curso
Ingeniería Informática	OB	2

Profesor/a de contacto

Nombre : Xavier Otazu Porter

Correo electrónico : xavier.otazu@uab.cat

Equipo docente

Joan Protasio Ramírez

Zakariae Majjouty Aouad

Victoria Montenegro Ruiz

Emanuel Adolfo Vallejo Canizales

Daniel Soto Alvarez

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 Programación 1 y Programación 2 de primer curso, y la asignatura de Bases de Datos de segundo curso.

Objetivos

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 Agile/SCRUM como paradigma de desarrollo evolutivo más relevante.

Resultados de aprendizaje

- CM16 (Integrar técnicas y metodologías adecuadas en el desarrollo de una solución informática que permitan una correcta gestión del trabajo en equipos humanos multidisciplinares) Integrar técnicas y metodologías adecuadas en el desarrollo de una solución informática que permitan una correcta gestión del trabajo en equipos humanos multidisciplinares
- SM20 (Analizar las diferentes fases del proceso de desarrollo de software para garantizar la fiabilidad y calidad de aplicaciones informáticas, empezando por una captura adecuada de los requisitos que permita realizar un modelado de la solución, una programación eficiente y el diseño del conjunto de pruebas) Analizar las diferentes fases del proceso de desarrollo de software para garantizar la fiabilidad y calidad de aplicaciones informáticas, empezando por una captura adecuada de los requisitos que permita realizar un modelado de la solución, una programación eficiente y el diseño del conjunto de pruebas

Contenidos

BLOQUE 1. EL SOFTWARE Y LA INGENIERÍA DEL SOFTWARE

Tema 1. Principios de la ingeniería del software.

- Definición y objetivos de la ES. 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 prototipaje. Modelo evolutivo. Modelo en espiral.

BLOQUE 2. MODELADO DE SOFTWARE

Tema 2. 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 3. 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 4. 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 3. ADMINISTRACIÓN DE LA CALIDAD DEL SOFTWARE

Tema 5. Calidad del software.

- Introducción. Conceptos de calidad.
- Prueba del software. Modelos de prueba del software. Estrategias de prueba. Los problemas del mal diseño de software. Principios básicos del diseño de software OO. TDD: Test-Driven Development.
- Administración de la configuración del software.

BLOQUE 4. ADMINISTRACIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE

Tema 6. SCRUM.

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

Actividades formativas y Metodología

Título	Horas	ECTS	Resultados de aprendizaje
Tutorías	15	0,6	
Clases de teoría	30	1,2	
Preparación y estudio	30	1,2	
Entrevistas de seguimiento de la práctica	15	0,6	
Trabajo práctico	38	1,52	
Seminarios / Problemas	15	0,6	

La asignatura de Ingeniería del software tiene una fuerte componente aplicada. Por este motivo, después de introducir los conceptos teóricos fundamentales, la asignatura se desarrolla según una metodología ABP (Aprendizaje Basado en Problemas). De esta manera, el alumno adquirirá los conocimientos a partir de un caso práctico (problema guía) de dimensión real. El problema guía dirigirá el proceso de aprendizaje a lo largo del curso, de manera que el trabajo práctico o los seminarios se estructurarán en torno a este caso. 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). Una vez impartidos los contenidos teóricos, se dedicarán todas las sesiones 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 adquirir una visión real del rigor, planificación y la sistemática que requiere el desarrollo profesional del software. No se profundizará en los temas 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. Las sesiones de teoría asumirán que el alumnado sigue los materiales distribuidos a través del campus virtual, por lo que se orientarán a explicar los conceptos clave, y fomentar el debate crítico sobre los contenidos. Para hacer las clases más dinámicas, se suele distribuir materiales previos al alumnado para autoaprendizaje (videos, enlaces web, presentaciones, etc.) y se pide que se trabajen antes de la clase, por lo que la clase se dirige a debatir los conocimientos, resolver dudas, incidir en lo más esencial y trabajar ejemplos prácticos.

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 la otra 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. Dado que se alternarán sesiones de teoría, problemas y prácticas a lo largo del curso, para una mejor organización, el alumnado dispondrá a principio de curso a través del campus virtual de la programación de todas las sesiones.

SEGUIMIENTO TRABAJO PRÁCTICO

No habrá sesiones presenciales de laboratorio. Los/las estudiantes resolverán el problema guía de manera autónoma en grupos reducidos, y el seguimiento se hará durante las horas de clase en el horario regular. Las sesiones obligatorias de prácticas (6 sesiones de 2 horas durante el curso) consistirán en dos partes. Primero, cada equipo de prácticas presentará al resto de la clase el progreso de su trabajo (el encargado de hacer la presentación será la persona que haga de scrum master en aquella iteración). En la segunda parte de la sesión el/la tutor/a de prácticas tendrá una entrevista individual con cada equipo para hacer el seguimiento y la evaluación. Se sigue un desarrollo basado en SCRUM, donde cada incremento, llamado sprint, corresponde a una sesión. La asistencia a las entrevistas de seguimiento será OBLIGATORIA para todos los alumnos/as. La no presencia incidirá negativamente en la evaluación del trabajo práctico, comportando no superar la evaluación si el número de no presencias supera el 15% de las sesiones. En caso de ausencia justificada, el profesorado considerará mecanismos de compensación. El tamaño de los equipos de prácticas se determinará a principio de curso. De manera orientativa serán equipos de 5 personas. Dado que la asistencia a las sesiones de seguimiento es obligatoria, y se hará durante los horarios presenciales, es importante que todas las personas miembros de un equipo de prácticas estén matriculados y asistan al mismo grupo.

TRABAJO PRÁCTICO

Se entregará a los alumnos al inicio de curso un problema guía. Este problema tendrá dos funciones. En primer lugar utilizarlo como ejemplo en las clases de teoría y seminarios, resolviendo partes adecuadas a los contenidos de cada momento. En segundo lugar se puede entender como el enunciado de una práctica que se va resolviendo por etapas a lo largo del curso. Los alumnos/as se organizarán en equipos de trabajo de 5

personas (según número de matriculados). La resolución del trabajo la harán de manera autónoma, con el seguimiento semanal antes descrito.

CLASES ONLINE

Aunque se espera que toda la docencia se pueda desarrollar en modalidad presencial, las sesiones de clase pueden tener que adaptarse a modalidad on-line a través de plataformas de videoconferencia. Cualquier modificación que afecte el formato y horarios, se informará a través del campus virtual y al inicio de curso.

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 (trabajo práctico)	50	4	0,16	CM16, SM20
Evaluación individual (examen de teoría)	50	3	0,12	CM16, SM20

Esta asignatura no contempla 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. En el caso de que la nota de algún parcial sea inferior a 5, se tendrá que recuperar en el examen de recuperación. La nota correspondiente a cada parcial tiene un peso del 50% en la nota final de la evaluación individual.

Por lo tanto, $AI = 50\% \text{ Nota_Parcial1} + 50\% \text{ Nota_Parcial2}$.

- Evaluación colectiva 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 (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.

La nota de la práctica es $AP = 60\% AC + 40\% AI$

INDICADORES Y VALORACIÓN:

La calificación final se obtendrá según el siguiente baremo:

QF = 50% AI + 50% AP

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 AC).

- 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 el valor menor entre 3.0 y la media ponderada de las notas 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 hora de 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" sobre medidas en casos de irregularidades por plagio en las actividades de evaluación.

ACTIVIDADES VOLUNTARIAS DE EVALUACIÓN:

Durante el curso se propondrán al alumnado actividades de entrega opcional que servirán para complementar las notas de las actividades obligatorias.

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.
- 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 en el campus virtual y pueden estar sujetos a cambios de programación por motivos de adaptación a posibles incidencias. Siempre informará al 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 notas 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. Esta información se anunciará el día de la presentación de la asignatura, y el campus virtual.

PLAGIO:

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). 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 aprobar la asignatura, esta asignatura quedará suspendida directamente, 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;
- tener dispositivos de comunicación (como teléfonos móviles, SmartWatch, etc.) accesibles durante las pruebas de evaluación teórico o Prácticas individuales (exámenes).

En resumen: copiar, dejar copiar o plagiar en cualquiera de las actividades de evaluación equivale a un SUSPENSO en toda la asignatura con nota inferior a 3,0.

En esta asignatura, no se permite el uso de tecnologías de Inteligencia Artificial (IA) en ninguna de sus fases. Cualquier trabajo que incluya fragmentos generados con IA será considerado una falta de honestidad académica y puede acarrear una penalización parcial o total en la nota de la actividad, o sanciones mayores en casos de gravedad.

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

Los contenidos de la asignatura explicados en clase se extraen de diferentes fuentes. Muchos materiales son en línea, extraídos de materiales multimedia, vídeos, etc. Durante el curso, se proporcionarán los materiales de confección propia necesarios para el seguimiento de la asignatura a través del campus virtual. También se proporcionarán enlaces a documentación en línea, software para los ejercicios prácticos, etc. en formato libre. A continuación se referencian algunos libros clásicos de ingeniería del software. Se recomienda al alumnado seguir la asignatura con los materiales actualizados que se recomendarán a clase.

- 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. Mc Graw-Hill, 7a. edición, 2010.
- A. Álvarez García, R. de las Heras del Dedo, C. Lasa Gómez, Métodos Ágiles. Scrum, Kanban, Lean. Anaya Multimedia, 2017.
- K. Pohl, C. Rupp. Requirements Engineering Fundamentals. Rocky Nook Inc. 2011.
- Alexey Krivitsky. Lego4Scrum. <https://www.lego4scrum.com/>

Software

Usaremos el Campus Virtual como herramienta de comunicación con los alumnos

- Scrum. Azure DevOps / Jira, Trello
- UML. PlantUML
- Gestión de la configuración. GitHub

Grupos e idiomas de la asignatura

La información proporcionada es provisional hasta el 30 de noviembre. 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

Tipo de docencia	Grupo	Idioma	Semestre	Turno
(PAUL) Prácticas de aula	411	Catalán/Español	segundo cuatrimestre	mañana-mixto
(PAUL) Prácticas de aula	412	Catalán/Español	segundo cuatrimestre	mañana-mixto
(PAUL) Prácticas de aula	431	Catalán/Español	segundo cuatrimestre	mañana-mixto
(PAUL) Prácticas de aula	432	Catalán/Español	segundo cuatrimestre	mañana-mixto
(PAUL) Prácticas de aula	451	Catalán/Español	segundo cuatrimestre	tarde
(PAUL) Prácticas de aula	452	Catalán/Español	segundo cuatrimestre	tarde