Universitat Autònoma de Barcelona

2022/2023

Diseño de Software

Código: 102759
Créditos ECTS: 6

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

Contacto

Nombre: Joan Serrat Gual
Correo electrónico: joan.serrat@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

Otras observaciones sobre los idiomas

Todos los materiales del curso (transparencias, enunciados) estan en inglés

Equipo docente

David Fernández Mota
Helena Bolta Torrell

Equipo docente externo a la UAB

Rodolfo Guichón

Prerequisitos

No hay prerrequisitos oficiales, pero es necesario que el estudiante domine la notación de diagramas de clase UML, los conceptos de orientación a objeto y su programación en algún lenguaje, preferiblemente Java.

Objetivos y contextualización

Una vez el propósito de un software se ha determinado mediante el análisis de requerimientos, los desarrolladores de software diseñan una solución. Esta va desde la arquitectura de sus componentes (clases, módulos ...) hasta la implementación en algún lenguaje de programación. Esta asignatura cubre principalmente dos aspectos claves del diseño: los patrones de diseño orientado a objeto y el diseño de la interfaz de usuario. Además de estos, trata otros temas como son el diseño por contrato de clases y métodos y el estilo de codificación.

El aprendizaje se apoya en un proyecto (la práctica) o dicho de otro modo, se trata de 'aprender para hacer': todos los conceptos tratados en teoría son necesarios para diseñar e implementar una aplicación de software, a partir de un enunciado que hace el papel de documento de requerimientos. Se trata de hacer el diseño y la

implementación (pero no el análisis de requerimientos, la prueba, el control de versiones o la garantía de la calidad, que pertenecen a otras asignaturas de la mención) de una aplicación.

Competencias

Ingeniería Informática

- Actitud personal.
- Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.
- Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.
- 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 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.
- Trabajar en equipo.

Resultados de aprendizaje

1. Asumir y respetar el rol de los diversos miembros del equipo, así como los distintos niveles de dependencia del mismo.
2. Concebir sistemas software basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.
3. Conocer y aplicar los componentes de software específicos en problemas computacionales basados en comunicaciones.
4. Desarrollar la curiosidad y la creatividad.
5. Diseñar una interface de usuario partiendo de especificaciones del cliente.
6. Diseñar y documentar soluciones software.
7. Especificar las necesidades del cliente en un documento de especificación de software.
8. Identificar el paradigma que mejor se adapta al problema de diseño de un sistema informático software específico teniendo en cuenta los requerimientos de accesibilidad, ergonomía, usabilidad y seguridad.
9. Identificar, gestionar y resolver conflictos.
10. Mantener una actitud proactiva y dinámica respecto al desarrollo de la propia carrera profesional, el crecimiento personal y la formación continuada. Espíritu de superación.
11. Trabajar cooperativamente.

Contenido

1. Principios de orientación a objeto
2. Patrones de diseño: creacionales, estructurales, de comportamiento
3. Diseño por contrato
4. Estilo de codificación y comentarios. logging
5. Diseño de la interfaz de usuario
- usabilidad
- UX research
- UX design : prototipado, user testing
- internacionalización y localización
6. Programación de interfaces en Flutter

Metodología

Clases de teoría. En ellas el profesor introduce los contenidos teóricos y da referencias útiles (libros, artículos, páginas web) para que el estudiante después pueda continuar su aprendizaje, así como desarrollar la aplicación software propuesta. Las transparencias utilizadas en estas clases pues no se deben tomar como fuente de estudio, sino que sirven para que el profesor pueda exponer de manera más ágil. En cambio, hay que estudiar los libros recomendados, y los artículos, capítulos de libros de acceso abierto y otros que ya indicaremos.

Clases de problemas. Estas clases de problemas se usaran principalmente para exponer ejercicios, que son problemas de diseño más implementación, o bien la aplicación de técnicas de UX. Dependiendo del calendario de cada curso, también se realizaran clases de problemas de la parte de patrones.

Sesiones de prácticas. Los estudiantes deberán formado a principio de curso grupos pequeños. En las clases de prácticas los estudiantes vienen a 1) mostrar el trabajo al professor y resolver dudas, y 2) dado que la práctica estará dividida en hitos, las semanas que corresponden a un hito evalúa la parte correspondiente in situ, según un baremo que figurará en el enunciado de la práctica. Por tanto, las sesiones de prácticas *no* son para hacer la práctica.

Totos els materiales de la asignatura, junto con una planificación detallada se encuentran en el campus virtual.

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 problemas	12	0,48	2, 3, 6, 5
clases de prácticas	12	0,48	1, 2, 3, 6, 5, 9, 11
clases de teoria	26	1,04	2, 3, 6, 5, 8
Tipo: Autónomas			
estudio individual	29	1,16	3, 6, 5, 7, 8
preparación de prácticas	52	2,08	1, 2, 3, 6, 5, 8, 9, 11
resolución de problemas	12	0,48	2, 3, 6, 5, 7, 8

Evaluación

La forma de avaluación es continua. Esta consta de una parte de evaluación de grupo y otra individual. La nota de la primera (PR) se obtiene del promedio de las notes de los hitos de práctiques (M1, M2, M3). La segunda (EX) del promedio de doss pruebas escrites durante el trimestre (EX1, EX2), o bien de un examen final de recuperación (EXR1, EXR2). Los sufijos 1 i 2 se refieren a las 2 partes que tiene la asignatura, antes y despues del primer examen EX1.

El algoritmo para el cálculo de la nota final (FM) es el siguiente (todas las notas son sobre 10) :

// EX1, EX2 examenes parciales
// EXR1, EXR2 examen de recuperación
// M1, M2, M3 notas de los 3 hitos de prácticas

```
// FM nota final

EX = ( max(EX1, EXR1) + max(EX2, EXR2) ) / 2.

if (M1>0) and (M2>0) and (M3>0) {
PR = (M1 + M2 + M3) / 3.
} else {
PR = 0.
}

if (EX >=5.) and (PR >=5.) {
FM =(EX + PR) / 2.
} else { // EX<5. o PR<5.
FM = min(EX, PR)
}
```

La práctica

La realización de la práctica és esencial para asimilar los conceptos de teoria. La primera cosa pues que queremos asegurar es que la pràctica haya sido realmente hecha por el grupo que la presenta, y que cada miembro del grupo haya contribuido en la misma medida que los demás. La segunda es que la práctica se haga de manera progresiva durante el curso, en paralelo a la teoria. Por eso haremos los siguiente:

- La nota de la práctica se obtiene combinando la nota de les partes en las que ésta se dividide. Cada parte será evaluada in situ en una de les sesiones de prácticas que anunciaremos a principio de curs ("hitos"). Los puntos que se pueden obtener de cada parte s'especificaran en el enunciado. Estas avaluaciones no tienen recuperación después.

- Los dias de avaluació de la práctica (hitos), si de resultas de les respuetas a las preguntas que haga el profesor durante la evaluación, tenemos dudas sobre la autoria o la contribución de alguno de los miembros del grupo, entonces la nota de práctica pasará de ser de grupo a individual y en función de cuanta i qué parte de la práctica haya hecho cada miembro.

Los ejercicios

Propondremos algunos ejercicios y pequeños problemas que son opcionales. Estos ejercicios se pueden hacer indivualmente o en pequeño grup, el mismo o diferente que el de práctiques. Los ejercicios pueden sumar punts a la nota de teoria EX, hasta un máximo de 10. Decimos 'pueden' porque la obtenciób de estos puntos extra estará condicionada a una prueba de validación para comprobar que el estudiant ha realitzado efectivamente exercicio que ha entregado. Estas pruebas de validación las haremos sólo una vez, en los exámenes EX1 y EX2, y no tienen recuperación después.

Otros

No convalidamos prácticas, exercicios ni exámenes de cursos anteriores. Los estudiantes repetidores no reciben ningún tratamiento diferenciado.

Si no se entrega un examen o una parte de la práctica, la nota de esta parte es cero. La nota final será 'no evaluable' si no se ha entregado nada evaluable, ya sea examen, práctica o ejercicio.

Los hitos de práctiques son evaluaciones equivalentes a examenes escritos. Si un estudiant no se presenta, la nota és cero.

Las notas MH las concederemos de manera discrecional a los estudiantes con nota final superior a 9.0 y teniendoen cuenta todo tipo de trabajos hechos (exámenes, práctica y ejercicios).

Las fechas de evaluación continuada y entrega de trabajos se publicaran en el campus virtual de la assignatura y pueden estar sujectas a cambios en respuesta a posibles incidéncies. Siempre se informará mediante el campus virtual, que será el mecanismo de comunicación entre docentes y estudiantes.

Para las actividades de evaluación de tipo examen escrito, se indicará un lugar, fecha y hora de revisión en la que el estudiantepodrá revisar la activitat con el professor. Si el estudiante no se presenta a esta revisión, no se revisará posteriorment esta actividad.

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 el estudiante que puedan conducir a una variación de la calificación de un acto de evaluación. Por lo tanto, plagiar, copiar o dejar copiar la 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
evaluación de grupo	50%	4	0,16	2, 3, 4, 6, 5, 7, 8, 10
evaluación individual	50%	3	0,12	1, 2, 3, 4, 6, 5, 7, 8, 9, 10, 11

Bibliografía

Part 1

- Design patterns: elements of reusable object oriented software. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Addison Wesley, 1994. Versión en castellano de Pearson Educación, 2003.
- Design patterns explained simply. Alexander Shvets. https://sourcemaking.com/design-patterns-ebook. También del mismo autor, https://refactoring.guru/design-patterns/
- Head first design patterns. E. Freeman, E. Freeman, K. Sierra, B. Bates. O'Reilly, 2004. Segunda edicion 2020.
- Clean code: a handbook of agile software craftmanship. R.C. Martin. Prentice Hall, 2008. Edición en castellano Codigo limpio, d'Anaya Multimedia 2012.

Part 2

- The user experience team of one. A research and design survival guide. Leah Buley. Rosenfeld, 2013.
- Validating product ideas. Tomer Sharon. Rosenfeld, 2016.
- Rocket surgery made easy: the do-it-yourself guide to finding and fixing usability problems. S. Krug. New Riders, 2010. Edición en castellano Haz facil lo imposible, d'Anaya Multimedia 2010.

Software

Lenguajes de programación: Java, Dart (Flutter)
Entornos de programación: IntelliJ IDEA (Java, Flutter), plugins Checkstyle, plantUML. Libreria logback
Otros :
- herramientas de prototipado de d'aplicacions para movil como marvelApp, inVision, Adobe XD etc.
- OptimalWorkshop para realitzar sesiones de user-testing y otras técnicas de UX