

DISEÑO DEL SOFTWARE

Curso 2012/2013

(Código:71013035)

1. PRESENTACIÓN DE LA ASIGNATURA

Programar una aplicación sencilla es fácil y puede ser incluso divertido. Pero la fabricación profesional de Software exige productos de funcionalidad generalmente sofisticada, con prestaciones y valores cualitativos que lo provean de ventajas competitivas en el mercado —lo que se denomina objetivos de negocio—. Y hacerlo requiere un buen número de destrezas y habilidades adicionales.

Ningún recurso es ilimitado y todos tienen un coste. Las tecnologías de desarrollo que se usen en la fabricación —y cómo se utilicen— repercuten en el coste. Precisamente hacia esto se enfoca la Ingeniería —en concreto, la *Ingeniería del Producto Software*—, que incorpora paradigmas en la producción como modularidad, flexibilidad, mantenibilidad, reusabilidad, desacoplamiento o genericidad. Porque no hay que olvidar que los objetivos de negocio se traducen en un balance entre las prestaciones del producto —su funcionalidad, sus restricciones y sus atributos, cualitativos o no— y sus costes.

En cualquier modelo de ciclo de vida, la secuencia de construcción incluye el trinomio Análisis y definición de requisitos, Diseño y Codificación. La asignatura se centra en el Diseño, donde se definen las '*piezas*' y los mecanismos de funcionamiento —el cómo—, que van a implementar la totalidad de las prestaciones del producto. Buena parte de las características que se exigen actualmente al Software aconsejan —para que los costes sean razonables— afrontar el desarrollo utilizando tecnologías que se basen en la orientación a objetos. Por tanto, el Diseño significa definir elementos de código —las clases—, lo que hacen, cómo lo hacen y cómo colaboran unas partes con otras.

Si hubiera que definir una destreza única para esta asignatura, se reduciría a 'Asignar Responsabilidades a los Componentes del Software'. El mayor nivel de abstracción en la Arquitectura la hace más adecuada para incorporar y verificar los atributos cualitativos.

El Diseño no se puede realizar sin el Análisis y tampoco estudiar el uno sin el otro. Mediante los Casos de Uso y los Diagramas de Secuencia, la OO establece unas pautas bien definidas para derivar los requisitos, que serán las '*cerchas*' para la construcción del Diseño Detallado. Pero aún quedan muchas preguntas sin resolver en el Diseño: ¿Qué componentes definimos? ¿Cómo les asignamos responsabilidades? ¿Por dónde empezamos, por la Arquitectura o por el Diseño Detallado? ¿Como incorporamos en el sistema los requisitos no funcionales —los atributos cualitativos— para alcanzar los objetivos de negocio?

El modelo de ciclo de vida iterativo se adecúa mucho mejor al comportamiento real del proceso de fabricación y favorece el refinamiento progresivo, que ayuda a paliar algunos de los problemas anteriores. Es el modelo que utiliza la asignatura, una simplificación del Proceso Unificado, que establece un camino bien definido para eliminar la incertidumbre del estudiante sobre qué debe hacer en cada instante del desarrollo.

Para resolver el objetivo principal —qué componentes definimos y cómo asignamos sus responsabilidades—, el núcleo de acero de la asignatura es el uso de los patrones de diseño. Un patrón es una solución contrastada y comprobada con éxito para una familia de problemas, y a la que se ha llegado tras catalogar un buen número situaciones y '*destilar*' las soluciones en un único resultado abstracto. La asignatura no está enfocada al estudio profundo del catálogo de patrones ni a su aplicación —un libro de referencia es "*Design Patterns*", de Gamma, Helm, Johnson y Vlissides; la llamada '*Pandilla de los Cuatro*', GoF—. La andadura controlada del Proceso Unificado permite introducir los patrones progresivamente, pero sólo a partir del punto en que tengamos la información organizada de tal forma que ya resulte más fácil asignar responsabilidades mediante los '*Principios Generales para Asignar Responsabilidades*' —patrones GRASP: *General Responsibility Assignment Software Pattern*—. Los patrones GRASP enseñan a asignar responsabilidades de manera que el diseño incorpore las especificaciones funcionales y algunos de los atributos cualitativos deseados. Tras los patrones GRASP, los patrones GoF más comunes enriquecen el diseño con los atributos que faltaban y favorecen el refinamiento de los que ya estaban. Nótese que esta manera de trabajar construye, de manera simultánea y coherente, el Diseño Detallado y la Arquitectura. Si se aplican



correctamente los patrones, se tienen garantías de que el Diseño Arquitectónico incluye los atributos cualitativos deseados, se puede evaluar y refinar para mejorar el comportamiento del sistema y el cumplimiento de los objetivos de negocio.

2.CONTEXTUALIZACIÓN EN EL PLAN DE ESTUDIOS

Esta asignatura forma parte de la materia de Ingeniería de Software —con 18 ECTS—, tiene carácter obligatorio y se sitúa en el quinto semestre del Grado en Ingeniería Informática. Su contribución al perfil profesional del Título está directamente vinculado al calificativo '*Ingeniería*' de su nombre y, con ello, su incidencia en las competencias, destrezas y habilidades relacionadas, simultáneamente, con la Ingeniería y el Desarrollo de Software.

Es el colofón para las asignaturas fundamentales orientadas al Desarrollo de Software, cuya trayectoria se inicia, en los primeros cursos, con las asignaturas relacionadas con los fundamentos de programación, las estructuras de datos y la algoritmia. Sus contenidos son más de tipo fundamental y metodológico que tecnológico; aunque sus enseñanzas estén estrechamente relacionadas y cimentadas con otras asignaturas que sí lo son —como 71901014 - Fundamentos de Sistemas Digitales; 71901066, 71902025 y 71012018 - Ingeniería de Computadores I, II y III; etc.—.

El antecedente inmediato de esta asignatura es 71902077 - Introducción a la Ingeniería de Software, en la que se accede a la antesala de las tecnologías de fabricación del Software como un producto comercial. El Diseño es una de las fases del desarrollo y es contigua a la codificación, la más directamente relacionada con lo que entendemos como Programación. Sin embargo, en este ámbito, la codificación se une a la intención ingenieril para convertirse en tecnologías que pretenden obtener del producto valores y cualidades adicionales que lo hagan competitivo. Por ello, es muy interesante relacionar esta asignatura con los conceptos de producción y objetivos empresariales a través de 71902031 - Gestión de Empresas Informáticas.

En lo que respecta a la Programación, el estudio del Diseño como vía para obtener ventajas y mejoras en los productos, no se puede abordar sin los paradigmas de la Orientación a Objetos, como ya se ha argumentado en la presentación. Por eso, la asignatura 71901072 - Programación Orientada a Objetos, es una referencia obligada.

Por todo ello, la asignatura incide en las siguientes competencias:

- Genéricas y Transversales (comunes a los grados de la UNED):
 - Competencias de gestión y planificación: Iniciativa y motivación. Planificación, coordinación y organización (establecimiento de objetivos y prioridades, secuenciación y organización del trabajo y el tiempo de realización, etc.).
 - Competencias cognitivas superiores: selección y manejo adecuado de conocimientos, recursos y estrategias cognitivas de nivel superior apropiados para el afrontar y resolver diversos tipos de problemas planteados en el desarrollo de software, con distinto nivel de complejidad y novedad:
 - Análisis y Síntesis (Diseño).
 - Aplicación de los conocimientos a la práctica.
 - Proporcionar una solución software a los problemas planteados en cualquier ámbito de negocio.
 - Pensamiento creativo.
 - Razonamiento crítico.
 - Toma de decisiones en función de objetivos.
 - Competencias de gestión de la calidad y la innovación: Seguimiento y evaluación del trabajo. Aplicación de medidas de mejora. Innovación.
 - Competencias de expresión y comunicación (a través de distintos medios y con distinto tipo de interlocutores): Comunicación y expresión matemática, científica y tecnológica.
 - Competencias en el uso de las herramientas y recursos de la Sociedad del Conocimiento: Manejo de las TIC. Competencia en la búsqueda de información relevante. Competencia en la recolección, gestión y organización de la información y los datos.



- Trabajo colaborativo:
 - Habilidad para coordinarse con el trabajo de otros.
 - Habilidad para negociar de forma eficaz.
 - Habilidad para la mediación y resolución de conflictos.
 - Habilidad para coordinar grupos de trabajo.
 - Liderazgo.
- Compromiso ético. Se fomentan actitudes y valores éticos, especialmente vinculados a un desempeño profesional ético. En un entorno competitivo, se previene contra la utilización ilícita de materiales y tecnologías con derechos de propiedad; animando a las soluciones originales o al planteamiento de la subcontratación.
- Específicas:
 - Capacidad para diseñar, desarrollar, seleccionar y evaluar, aplicaciones software, asegurando su fiabilidad, seguridad y calidad, conforme a los principios éticos y a la legislación y normativa vigente.
 - Capacidad para implantar y peritar proyectos, servicios y sistemas software, liderando su puesta en marcha y mejora continua y valorando su impacto económico y social.
 - Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en entornos de desarrollo de software.
 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente.
 - Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.
 - 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.
 - Capacidad para 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.

Las competencias anteriores son complejas. En algunos casos, la asignatura matiza y complementa aspectos que ya se han desarrollado parcialmente en las asignaturas mencionadas más arriba. En otros casos, aporta y desarrolla completamente facetas concretas de cada competencia; en particular las que se refieren al análisis y diseño orientados al desarrollo robusto y eficiente de productos de calidad, a su mejora y valoración para su implantación o peritación.

En la frontera superior se encuentran otras asignaturas, obligatorias y optativas, con carácter tecnológico e ingenieril. Pero es en el octavo semestre donde se culmina la materia de Ingeniería de Software con la asignatura 71014052 - Gestión de Proyectos Informáticos, con una visión integradora desde la gestión de todas las actividades.

3.REQUISITOS PREVIOS REQUERIDOS PARA CURSAR LA ASIGNATURA

La formación previa que deberían tener los alumnos para el adecuado seguimiento de esta asignatura es la propia del segundo curso de este Grado. En concreto:

- Conocimiento y experiencia en algún lenguaje de programación orientado a objetos; en particular, Java.
- Conocimiento de los conceptos generales de la tecnología de objetos: *clase, instancia, interfaz, abstracción, polimorfismo, encapsulación y herencia*.

Estos conocimientos y destrezas se pueden adquirir mediante las asignaturas "71901020 - Fundamentos de Programación" y "71901072 - Programación Orientada a Objetos".



Para poder entender y aprovechar la acusada intención ingenieril de esta asignatura, es muy recomendable conocer los principios fundamentales de la *Ingeniería de Software*. La asignatura "71902077 - Introducción a la Ingeniería de Software" proporciona la formación necesaria para asimilar los aspectos de *Ingeniería del Desarrollo de Software* que se tratan en esta asignatura.

4.RESULTADOS DE APRENDIZAJE

Con carácter general, el estudiante debe adquirir el conocimiento necesario para pasar desde el desarrollo de aplicaciones sencillas, a partir de especificaciones funcionales —u otras de tipo técnico—, a saber cómo incorporar atributos cualitativos que le permitan construir software robusto, mantenible, de altas prestaciones, calidad competitiva y costes razonables. Como resultado del aprendizaje de los contenidos de esta asignatura, el alumnado estará en la situación siguiente:

- Conoce la Ingeniería de Requisitos y las técnicas específicas de diseño de software enfocadas al desarrollo orientado a objetos.
- Establece la relación entre el diseño software y la facilidad con que se alcanzan las restricciones técnicas y atributos cualitativos del sistema. Conoce qué es y cuáles son los estilos arquitectónicos más importantes.
- Relaciona el diseño software con los objetivos de negocio de la organización. Identificar las restricciones y atributos de calidad que debe cumplir un sistema. Construye una arquitectura en función de los requisitos cualitativos que debe cumplir un sistema sencillo.
- Representa un diseño en relación a los requisitos cualitativos y las restricciones que debe cumplir.
- Analiza, valora y razona las ventajas e inconvenientes de un diseño utilizando criterios técnicos, cualitativos y del dominio del negocio.
- Es capaz de debatir y defender las conclusiones con argumentos sólidos y bien fundados.
- Conoce y sabe aplicar herramientas de soporte orientadas a evaluar los aspectos cuantitativos y económicos para tomar decisiones, en el desarrollo de un diseño, que se conjuguen adecuadamente con los objetivos de negocio de la organización.
- Aplica las técnicas de optimización aprendidas para el diseño arquitectónico, en arquitecturas que gestionen la variabilidad y mejoren la flexibilidad de la producción y la competitividad de la organización.

Los resultados del aprendizaje anteriores, se traducen en conocimientos (qué va a conocer o a saber), destrezas y habilidades (qué va a saber manejar o hacer) y aptitudes (para qué estará capacitado), que se adquieren una vez superada la asignatura.

1. Conocimientos

Adquirirá —y deberá demostrar— los siguientes conocimientos:

- Conocer y saber aplicar los principios y metodologías del ciclo de vida evolutivo.
- Saber identificar los requisitos, restricciones y atributos de calidad que debe cumplir un sistema.
- Saber relacionar el diseño software con la especificación funcional y los objetivos de negocio de la organización.
- Saber construir un diseño en función de los distintos tipos de requisitos y atributos de un sistema sencillo.
- Saber qué son los patrones de diseño y su utilidad.
- Saber los 9 principios fundamentales en el diseño de objetos y asignación de responsabilidades. Saber cómo aplicarlos.
- Conocer algunos patrones GoF y saber qué mejoras proporcionan al diseño.
- Conocer qué es y cuáles son los estilos arquitectónicos más importantes. Saber establecer la relación entre una arquitectura software y la facilidad con que se alcanzan las restricciones técnicas y atributos cualitativos del sistema.

2. Destrezas



Tras el estudio de la asignatura el alumno deberá:

- Saber utilizar la notación de UML.
- Identificar Casos de Uso primarios y registrarlos de forma completa en el estilo esencial.
- Identificar las clases conceptuales, asociaciones y sus atributos en el dominio del problema y recogerlas en un modelo conceptual mediante la notación UML.
- Identificar eventos del sistema y representarlos en un diagrama de secuencia mediante UML.
- Asignar responsabilidades y diseñar colaboraciones entre los componentes del Software mediante patrones GRASP e ilustrar el resultado mediante diagramas de colaboración en UML.
- Resolver las definiciones de clases en un diagrama de clases en UML.
- Refinar el diseño y mejorar la arquitectura mediante la incorporación de patrones GoF.
- Hacer corresponder los artefactos de diseño con definiciones de clases en un lenguaje de programación orientada a objetos.
- Analizar, valorar y razonar las ventajas e inconvenientes de un diseño arquitectónico utilizando criterios técnicos y cualitativos del dominio de negocio.
- Habilidad para debatir y defender las conclusiones con argumentos sólidos y bien fundados.

3. Aptitudes

Una vez superada la asignatura el alumno estará capacitado para:

- Analizar, diseñar y construir aplicaciones de forma robusta, segura y eficiente, eligiendo las tecnologías de desarrollo más adecuadas.
- Desarrollar, mantener y evaluar servicios y sistemas software que satisfagan los requisitos del usuario, las necesidades del cliente y los objetivos del dominio de negocio.
- 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.

5.CONTENIDOS DE LA ASIGNATURA

El programa de la asignatura se estructura en tres unidades didácticas y recoge cinco temas que se corresponden con los 34 primeros capítulos del texto base. Este programa realiza un recorrido, gradual y detallado, por las actividades y los aspectos significativos del diseño y desarrollo de cualquier producto software. La misma estructura de los contenidos establece un método incremental de trabajo, válido para afrontar tanto el estudio de la asignatura como las actividades implicadas en la elaboración de software.

Unidad Didáctica I

1. Introducción.

Contenidos: El Análisis y Diseño Orientado a Objetos: objetivos. El Proceso Unificado y el desarrollo iterativo: método de trabajo. Sistema de punto de venta: escenario práctico de seguimiento.

2. Inicio.

Contenidos: Fase de Inicio. Comprensión de los requisitos. Modelo de Casos de Uso: escritura de los requisitos en el contexto. Identificación de otros requisitos. Transición a la elaboración.



Unidad Didáctica II

3. Primera iteración de la fase de elaboración.

Contenidos: Modelo de Casos de Uso: representación de los Diagramas de Secuencia. Modelo del Dominio: registro de conceptos, añadir asociaciones, añadir atributos y añadir detalles con los contratos de las operaciones. De los requisitos al diseño. Notación de los diagramas de iteración. GRASP: diseño de objetos con responsabilidades (patrones). Modelo de Diseño: elaboración de los Casos de Uso con patrones GRASP, determinación de la visibilidad, creación de los Diagramas de Clases de diseño. Modelo de Implementación: transformación de los diseños en código.

4. Segunda iteración de la fase de elaboración.

Contenidos: Requisitos de la iteración 2. Más patrones GRASP para asignar responsabilidades. Patrones GoF para diseñar las realizaciones de los Casos de Uso.

Unidad Didáctica III

5. Tercera iteración de la fase de elaboración.

Contenidos: Requisitos de la iteración 3. Relaciones entre Casos de Uso. Modelado de la Generalización. Refinamiento del Modelo de Dominio. Ampliación de DSSs y contratos. Modelado del comportamiento con Diagramas de Estado. Diseño de la Arquitectura lógica con patrones. Organización de los paquetes de los Modelos de Diseño e Implementación. Introducción al Análisis de Arquitecturas y el SAD. Diseño de más realizaciones de Casos de Uso con objetos y patrones. Diseño de un 'marco de trabajo' (*framework*) de persistencia con patrones.

6.EQUIPO DOCENTE

- [JOSE FELIX ESTIVARIZ LOPEZ](#)
- [JAVIER ARELLANO ALAMEDA](#)

7.METODOLOGÍA Y ACTIVIDADES DE APRENDIZAJE

La metodología que se usa en la enseñanza de la asignatura es la propia de la UNED y está basada en el aprendizaje a distancia ayudado por la infraestructura, los recursos materiales de la Universidad y humanos a nuestro alcance y apoyado por el uso de las tecnologías de la información y el conocimiento.

En este apartado hemos de distinguir entre cómo aprenderá el alumno esta asignatura (actividades formativas) y con qué medios cuenta para llevar a cabo dicho aprendizaje.

1. Las actividades formativas para el estudio de la asignatura son:

- Estudio de los contenidos relativos a los procedimientos y tecnologías de desarrollo del programa: En esta actividad el alumno debe desarrollar un trabajo autónomo que consiste en el estudio de la materia utilizando el libro de texto básico ("UML y Patrones").
- Realización de ejercicios teórico-prácticos: Esta actividad consiste en la realización, por parte del alumno y de forma autónoma, de los desarrollos ilustrativos y ejemplos prácticos que utiliza el texto básico para mostrar cómo se ejecutan las actividades y los planteamientos conceptuales de las tecnologías de desarrollo. A este respecto, el texto de referencia utiliza un único escenario —Sistema de Punto de Venta (SPV)— para ilustrar los contenidos del programa.

2. Los materiales útiles y necesarios para el aprendizaje son: la Bibliografía Básica, el Curso Virtual y la página Web de la asignatura.



1. La Bibliografía Básica consta de:

- El texto básico que el alumno debe usar para el estudio teórico-práctico de la materia objeto de la asignatura. El programa de la asignatura se ajusta a los 34 primeros capítulos del libro, que está perfectamente adaptado al estudio autónomo. Cada tema del programa está correlacionado con un grupo de capítulos adyacentes en el libro y, cada uno de estos, incluye la contextualización de sus contenidos, los objetivos y resultados esperados de aprendizaje. Análogamente, en cada tema del programa, hay una colección de cuestiones para el repaso y la reflexión de sus contenidos, así como referencias para la lectura e información adicionales.

2. En el Curso Virtual de la asignatura el alumno encontrará:

- Un calendario con la distribución temporal de los temas a lo largo del cuatrimestre y con las fechas de entrega de las distintas actividades teórico-prácticas que el alumno puede realizar para su evaluación. Lógicamente la distribución del estudio de los temas es totalmente orientativa y no está obligado a seguirla salvo en lo referente a las fechas de entrega de las pruebas de evaluación a distancia calificables que, tras la fecha de entrega, se cerrará la aplicación informática y no serán posible entregarlas.
- Una guía de estudio en la que se hace una descripción detallada del plan de trabajo propuesto con el fin de orientar al alumno en el estudio de cada uno de los temas de la asignatura. Además, se especifican los conceptos y desarrollos más importantes, así como las habilidades y aptitudes que el alumno debe haber conseguido tras el estudio de dicho tema.
- La relación y descripción de las pruebas de evaluación a distancia calificables, y las normas y condiciones que deben tener en cuenta para la entrega de dichas actividades.
- El software de ayuda para modelado y representación gráfica [Astar/Community](#), concebido para usar UML.

Además, para establecer una comunicación fluida entre profesores y alumnos, dentro del Curso Virtual existen las siguientes facilidades:

- Los foros por medio de los cuales los profesores y/o tutores aclararán las dudas de carácter general y específicas de cada uno de los temas. También se usarán para comunicar todas aquellas novedades que surjan a lo largo del curso.
- Calendario de planificación y alertas, para la organización del estudiante.
- Comunicación mediante correo electrónico a través del servicio interno de la plataforma aLF.
- Área de entrega y calificación de las pruebas de evaluación calificables.
- Servicio de repositorio documental, para disponer de materiales y documentación adicionales.

3. En la [página Web de la asignatura](http://www.issi.uned.es/Grado/SWDes/index.htm) —<http://www.issi.uned.es/Grado/SWDes/index.htm>— el alumno encontrará:

- Toda la información de naturaleza estática existente en el Curso Virtual, incluida la presentación, Guía Didáctica de la asignatura, Plan de Trabajo, integrantes del Equipo Docente, direcciones de contacto, etc.
- Una relación y el acceso a los documentos con las actividades de autoevaluación y pruebas de evaluación calificables.
- El sistema de evaluación y calificación, con las normas y condiciones que deben tener en cuenta al evaluar los resultados del aprendizaje individuales.
- Las herramientas y utilidades de soporte y ayuda que resulten de interés para el seguimiento de la asignatura; cuando sea viable su distribución libre al alumnado, como [Astar/Community](#).
- Cualquier información, documento u otro elemento que sea de interés para la asignatura, se actualizará o añadirá en esta página, durante el transcurso del semestre, y en el Curso Virtual.

Además, para respaldar el soporte y la comunicación entre profesores y alumnos, se dispone:

- El correo electrónico swdesign@issi.uned.es mediante el cual, de forma privada, se puede comunicar con los profesores para plantear todas aquellas dudas o cuestiones académicas que se les presenten o consideren oportunas.
- La atención personal por parte del Equipo Docente, en el despacho, los días de guardia y por medio del teléfono para aquellos casos que lo requieran. Así mismo, los Tutores atienden personalmente a los alumnos en las tutorías que tienen lugar en los Centros Asociados.

8.EVALUACIÓN

En el proceso de aprendizaje es de gran relevancia poder evaluar tanto el grado y la calidad en que los conocimientos y destrezas exigidos en la asignatura han sido adquiridos por el estudiante como la manera en que se han transmitido y la evolución del alumnado en este proceso. Además de la evaluación o examen final, una manera adecuada de conseguirlo es



realizar una evaluación continua a lo largo del semestre, con el fin de informar a docentes y alumnado de la trayectoria del aprendizaje. Hay que hacer notar que no son dos tipos distintos de evaluación, sino dos formas que participan del propio proceso de maduración en la asignatura. Si bien el examen final tiene un carácter no-interactivo, determinista o finalista más acusado, la evaluación continua significa un excelente entrenamiento para la prueba final (que será similar) y provee de mecanismos muy valiosos para la auto-supervisión y seguimiento por parte del alumnado y los docentes. Cada forma de evaluación consta, a su vez, de varios apartados que cubren las distintas competencias, habilidades y destrezas que debe haber adquirido el estudiante una vez que ha superado dicha asignatura.

1. EVALUACIÓN CONTINUA.

Dentro de este tipo de evaluación consideramos dos modalidades: Actividades de Autoevaluación y Actividades Calificables. Esta evaluación la realiza el Profesor Tutor.

1. Ejercicios y Actividades de Autoevaluación.

La autoevaluación es de gran importancia en el proceso general de aprendizaje de la asignatura, ya que informa al alumno de la trayectoria que va llevando en la adquisición del conocimiento de la materia durante el proceso de aprendizaje autónomo. Para ello, al final de cada tema del programa, en el capítulo correspondiente del texto base, hay un epígrafe denominado "No se entiende «concepto» cuando...". Los epígrafes contienen una colección de cuestiones y planteamientos que el alumno debe saber responder tras haber estudiado y comprendido adecuadamente el tema correspondiente y sobre los que debe reflexionar para auto evaluarse o establecer otros puntos de vista en la comprensión de esos contenidos.

2. Actividades calificables por el Profesor Tutor.

Esta evaluación continua la realiza el Profesor Tutor del Centro Asociado —o el que corresponda en la virtualización—, mediante las Pruebas de Evaluación a Distancia. Consisten en una actividad cuya descripción e instrucciones para su realización y entrega están descritas con detalle en la Segunda Parte de la Guía de Estudio y en el material del curso virtual de la asignatura. Tendrá un esquema idéntico a la primera sección del examen final.

La actividad debe entregarse en la forma, la fecha y por el procedimiento que se indican en la Segunda Parte de la Guía de Estudio. Es importante tener en cuenta que no hay posibilidad de entregar ninguna actividad una vez superada la fecha indicada, porque se cerrará la aplicación y es imprescindible que se realice y evalúe en el semestre en el que se imparte la asignatura. Por tanto, en la convocatoria de septiembre se mantiene la nota obtenida en la actividad durante dicho semestre.

Como ya se ha comentado, son evaluadas por el Profesor Tutor del Centro Asociado correspondiente y la calificación obtenida se tienen en cuenta en la nota final de la asignatura de la siguiente forma:

- La Actividad de Evaluación a Distancia se puntúa sobre 10. Estará superada si su calificación es de 5 o mayor.
- Para que esta Nota obtenida en la Prueba de Evaluación a Distancia se tenga en cuenta en el cómputo de la Nota Final de la Asignatura es imprescindible cumplir las siguientes condiciones:
 - Haber aprobado la Actividad Calificable a Distancia (NPED $\geq$ 5).
 - Que la calificación obtenida en la Prueba de nivel Presencial o Examen Final sea mayor o igual a 4'5 (sobre 10).
- Si se cumplen las anteriores condiciones, la Nota Final de la Prueba de Evaluación a Distancia siempre es aditiva en el cómputo de la Nota Final de la Asignatura y la máxima contribución es de 2 puntos (sobre 10). Es decir, en el peor escenario, en el que se haya obtenido 4'5 en el Examen Final y la puntuación de la Actividad Calificable fuera de 5, la aportación de la Prueba de Evaluación a Distancia sería $5 \times 2 / 10 = 1$, valor que se sumaría a 4'5 para obtener la Nota Final de la Asignatura (aprobada). En el mejor caso, con un 10 en la PED, su aportación sería un 2, que se sumaría a la calificación del examen (si supera el 4'5).
- En el caso de que la calificación de la PED contribuya a la Nota Final de la Asignatura, su aportación será del 20% relativo a la calificación obtenida en la primera sección del Examen Final (NE1s será el 80%). Esto es aplicable sólo a partir del valor 6'25 obtenido en NE1s (6'25 $\leq$ NE1s, ver cuadro más abajo).

2. EVALUACIÓN FINAL.

La evaluación final se corresponde con el Examen Final o Prueba Presencial de la asignatura. Se realiza en los Centros Asociados, tiene una duración de 2 horas y para su ejecución no se permite el uso de calculadora ni material adicional.

La Prueba Presencial consiste en una batería de preguntas teórico/prácticas y ejercicios (alrededor de 8), formulados en torno a un escenario o el dominio de un problema concreto. En este tipo de asignaturas la solución de un problema no es única y existe un sinnúmero de variantes sobre una respuesta correcta estandarizada. Cada uno de los apartados del examen requiere que se construya la respuesta basándose en la respuesta del apartado anterior. Se puntúa sobre 10, pero la puntuación de los apartados no es la misma para todos: las preguntas relativas a los Casos de Uso tienen menor puntuación, ya que son relativamente inmediatas y su respuesta no está directamente relacionada con las habilidades del diseño orientación a objetos. Por el contrario, las cuestiones relacionadas con la asignación de responsabilidades, patrones y diseños de colaboración tienen las puntuaciones más altas.

Adicionalmente, la prueba tiene dos o tres cuestiones de carácter conceptual y de conclusiones sobre los contenidos de la asignatura, cuya máxima puntuación conjunta es de 1 punto. Esta sección se utiliza para elevar la calificación final de la asignatura (BP); siempre y cuando la puntuación de la otra sección del ejercicio sea mayor o igual que 4'5 y su suma con la contribución de la PED (CPED = NPED $\times$ 2 / 10) sea mayor o igual que 5 (es decir: 6'25 $\leq$ NE1s).



¡Atención! La puntuación mínima para superar el Examen Final es de 5. El cálculo para determinar los límites se hace, exclusivamente, con las puntuaciones obtenidas en la primera sección —la batería de preguntas teórico/prácticas—, más la contribución de la PED —si corresponde— y excluyendo la sección de cuestiones adicionales (BP). Solamente si la puntuación obtenida en la 1ª sección (NE1s) es mayor o igual a 6'25 y la PED está aprobada, los puntos adicionales (BP) se añaden a la nota final de la asignatura (NF)

La asignatura estará superada con una calificación igual o mayor que cinco (**$NF \geq 5$**). A continuación se incluye un cuadro-resumen con los algoritmos utilizados para la calificación de las diferentes partes de la evaluación.

Nota Examen Final			Nota PED		Nota Final Asignatura
Primera Sección (NE1s)	Puntos adicionales (BP)	Nota Examen (NPP)	Calificación PED (NPED)	Contribución PED (CPED)	
NE1s < 4'5	BP	NPP = NE1s	NPED	CPED = 0	NF = NPP
4'5 ≤ NE1s < 6'25	BP	NPP = NE1s + CPED	NPED < 5	CPED = 0	NF = NPP
			5 ≤ NPED	CPED = NPED x 2 / 10	
6'25 ≤ NE1s	BP	NPP = (NE1s x 0'8) + CPED	NPED < 5	CPED = 0	NF = NPP
			5 ≤ NPED	CPED = NPED x 2 / 10	NF = NPP + BP

9.BIBLIOGRAFÍA BÁSICA

ISBN(13): 9788420534381

Título: UML Y PATRONES. UNA INTRODUCCIÓN AL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS Y AL PROCESO UNIFICADO. SEGUNDA EDICIÓN (Segunda)

Autor/es: Larman, C. ;

Editorial: PEARSON EDUCACIÓN

Buscarlo en Editorial UNED

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

10.BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13): 9788478290598

Título: PATRONES DE DISEÑO

Autor/es: Gamma, Erich ; Otros ;

Editorial: PEARSON ADDISON-WESLEY

Buscarlo en librería virtual UNED



Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9789684443648

Título: UML GOTA A GOTA

Autor/es: Fowler, Martin ; Kendall, Scott ;

Editorial: PEARSON

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

11.RECURSOS DE APOYO

Para ayudar en el estudio de la asignatura, el estudiante dispondrá de diversos medios de apoyo. Así, podemos destacar los siguientes:

- El equipo docente que está a disposición de los estudiantes para orientarle y ayudarle en el aprendizaje de la asignatura y en la preparación de las pruebas presenciales.
- El curso virtual que constituye el principal punto de apoyo.
- Las tutorías presenciales en el Centro Asociado correspondiente y que son un gran apoyo para el estudiante. En ellas el profesor tutor resuelve dudas y cuestiones académicas y explica aquellos conceptos y/o temas que resultan más complicados.
- En la [página Web de la asignatura](http://www.issi.uned.es/Grado/SWDes/index.htm) —<http://www.issi.uned.es/Grado/SWDes/index.htm>— se pueden encontrar las herramientas y utilidades de soporte y ayuda que resulten de interés para su seguimiento; cuando sea viable su distribución libre al alumnado, como [Astah/Community](#).
- Existen también muchos recursos en Internet. No sólo información y documentación sobre POO, Diseño y patrones, sino herramientas para el modelado, CASE o entornos de programación OO. El equipo docente no recomienda que el estudiante desvíe su atención con la búsqueda, instalación, configuración y aprendizaje de herramientas adicionales. Es más aconsejable reflexionar, usar papel y lápiz. En última instancia, sí es recomendable codificar los diseños y comprobarlos con un entorno como [BlueJ](#) o [Eclipse](#).

12.TUTORIZACIÓN

1. Profesores Tutores (en el Centro Asociado correspondiente)

Los tutorías presenciales tienen lugar en los Centros Asociados. Son atendidas por los Profesores Tutores y sus horarios los pueden encontrar en dichos centros.

2. Equipo docente (en la Sede Central)



Las guardias tendrán lugar todos los lunes y jueves lectivos de 16h a 20h.

Los profesores que les atenderán son:

José Félix Estívariz López

Dpto. de Ingeniería de Software y Sistemas Informáticos. Despacho 2.20.
Lunes de 16h a 20h. Teléfono 913987792

Rubén Heradio Gil

Dpto. de Ingeniería de Software y Sistemas Informáticos. Despacho 2.20.
Jueves de 16h a 20h. Teléfono 913988242

Dirección:

ETS de Ingeniería Informática de la UNED
Dpto. de Ingeniería de Software y Sistemas Informáticos. Despacho 2.20.
C/ Juan del Rosal, 16
28040 – Madrid

El procedimiento recomendado para comunicar con los profesores es el correo electrónico dentro del Curso Virtual de la asignatura o bien mediante el teléfono en el horario de guardias y permanencias. También pueden usar el correo electrónico de la asignatura swdesign@issi.uned.es.

Si, al realizar una llamada por teléfono, los profesores del equipo docente no le pueden atender en ese momento, le recomendamos que dejen un mensaje en el contestador en el que es muy importante que deje bien claro su nombre completo, su número de teléfono y la asignatura objeto de la llamada, para que el equipo docente le pueda devolver la llamada.

