20-21

GRADO EN INGENIERÍA INFORMÁTICA
TERCER CURSO

GUÍA DE ESTUDIO PÚBLICA



DISEÑO DEL SOFTWARE

CÓDIGO 71013035

UNED

20-21

DISEÑO DEL SOFTWARE

CÓDIGO 71013035

ÍNDICE

PRESENTACIÓN Y CONTEXTUALIZACIÓN
REQUISITOS Y/O RECOMENDACIONES PARA CURSAR LA ASIGNATURA
EQUIPO DOCENTE
HORARIO DE ATENCIÓN AL ESTUDIANTE
TUTORIZACIÓN EN CENTROS ASOCIADOS
COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE
RESULTADOS DE APRENDIZAJE
CONTENIDOS
METODOLOGÍA
SISTEMA DE EVALUACIÓN
BIBLIOGRAFÍA BÁSICA
BIBLIOGRAFÍA COMPLEMENTARIA
RECURSOS DE APOYO Y WEBGRAFÍA

Nombre de la asignatura	DISEÑO DEL SOFTWARE
Código	71013035
Curso académico	2020/2021
Departamento	INGENIERÍA DEL SOFTW. Y SIST. INFORMÁTICOS
Título en que se imparte	GRADO EN INGENIERÍA INFORMÁTICA
CURSO - PERIODO	GRADUADO EN INGENIERÍA INFORMÁTICA - TERCER CURSO - SEMESTRE 1
CURSO - PERIODO	ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE SISTEMAS - OPTATIVAS CURSO - SEMESTRE 1
CURSO - PERIODO	ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE SISTEMAS EN UNED - OPTATIVAS CURSO - SEMESTRE 1
Tipo	OBLIGATORIAS
Nº ETCS	6
Horas	150.0
Idiomas en que se imparte	CASTELLANO

PRESENTACIÓN Y CONTEXTUALIZACIÓN

Presentación

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 él. 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. 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.

Etimológicamente, el término general '*diseño*' se refiere a la representación de la o las propuestas que dan solución a un reto o problema planteado (*designio*). Es decir, significa la representación de un propósito (de algo aún no realizado), pero que **requiere definir cómo** se construye y **de qué manera resuelve el problema**. De igual forma, el verbo '*diseñar*' es el proceso que conduce a esa representación, al diseño, e involucra acciones como la observación, la investigación, el análisis, el modelado y las pruebas, ajustes o adaptaciones previas a la producción definitiva. En ingeniería, **"diseño" es el conjunto de especificaciones técnicas que definen la solución propuesta**.

La asignatura se centra en el Diseño, donde se definen las '*piezas*' y los mecanismos de funcionamiento —**el cómo** va a funcionar la aplicación, cada una de sus partes, y **cómo** se

van a ensamblar para que funcione así—, las cuales 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.

Sin embargo, a no ser por las técnicas de modelado (y su representación), o por las relativas a cómo organizar las actividades en el proceso del diseño, las asignaturas precedentes a ésta capacitan completamente al estudiante para elaborar, con código, una solución que satisfaga los requisitos funcionales de un problema software (ver los programas de las asignaturas de programación, algoritmia, estructuras de datos, etc., que se referencian en el epígrafe de "Contextualización"). Quizás no sepa representarlo adecuadamente, pero ya lleva tiempo *haciendo diseños*. A partir de la asignatura Introducción a la Ingeniería de Software, esas capacidades para el manejo de los recursos de los lenguajes de programación convergen con un *proceso sistemático del desarrollo* y, sobre todo, con la valoración de ciertas cualidades de las soluciones buscadas: la independencia funcional, la adaptabilidad y la comprensibilidad de los diseños.

Por consiguiente, **es extremadamente importante que el estudiante comprenda que**, aunque esta asignatura incide en cómo organizar las actividades involucradas en el diseño, dentro de un proceso de desarrollo, y en cómo representar las conclusiones y artefactos obtenidos en ellas, no pretende detenerse en enseñar cómo obtener un diseño (algo para lo que el estudiante ya debe estar capacitado), sino que **su objetivo principal es enseñar cómo conseguir que la solución funcional propuesta (el diseño) sea funcionalmente independiente, adaptable y comprensible**. Todos los contenidos y materiales de estudio de esta asignatura están enfocados a este objetivo.

Si hubiera que definir una destreza única para esta asignatura, se reduciría a 'Asignar Responsabilidades a los Componentes del Software'. Un mayor nivel de abstracción, en la Arquitectura, la hace más adecuada para incorporar y verificar los atributos cualitativos que se quieren conseguir en el producto final. Pero éstos *no se inyectan* en los módulos de la arquitectura, sino que están imbricados en los elementos más atómicos de código que los constituyen (en el diseño detallado).

En cualquier modelo de ciclo de vida, la secuencia de construcción incluye el trinomio { Análisis y definición de requisitos —Diseño —Codificación }. El Diseño ni se puede realizar sin el Análisis ni tampoco estudiar el uno sin el otro. Mediante los Casos de Uso y el registro de la secuencia de las interacciones que se producen en cada uno de ellos (o los Diagramas de Secuencia del Sistema), 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 (funciones y métodos)? ¿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 responder alguna

de las preguntas 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 la utilización de unos **principios de diseño**. Mediante el uso y la aplicación de los '*Principios Generales para Asignar Responsabilidades*' —principios **GRASP**: ***G****eneral* ***R****esponsibility* ***A****ssignment* ***S*** *oftware* ***P****attern*—, en el ámbito controlado del Proceso Unificado, se consigue organizar la información del sistema en desarrollo de tal forma que ya resulte más fácil asignar responsabilidades. Únicamente a partir de este punto es posible introducir los patrones, progresivamente, en el diseño. Un patrón de diseño 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**—. El autor del libro de esta asignatura llama *patrones* a los principios GRASP por hacer una analogía con el concepto fundamental del patrón de diseño en su aplicación a una familia de problemas y situaciones bien conocida pero, cada GRASP, es un principio o directriz para realizar el diseño. Los principios GRASP enseñan a asignar responsabilidades de manera que el diseño incorpore las especificaciones funcionales y algunos de los atributos cualitativos deseados. Tras los principios 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 principios GRASP, se tienen garantías de que el Diseño Arquitectónico incluya los atributos cualitativos deseados, y se puede evaluar y refinar para mejorar tanto el comportamiento del sistema como el cumplimiento de los objetivos de negocio.

Contextualización

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 de empresa 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.
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.

REQUISITOS Y/O RECOMENDACIONES 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:

- Nivel avanzado en las competencias de **programación OO** y, en particular, de Java; FB.05: Conocimiento del funcionamiento de los sistemas informáticos, así como de su programación y su aplicación para la resolución de problemas propios de la ingeniería.
- 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.

EQUIPO DOCENTE

Nombre y Apellidos	JOSE FELIX ESTIVARIZ LOPEZ (Coordinador de asignatura)
Correo Electrónico	jose.estivariz@issi.uned.es
Teléfono	91398-7792
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	JAVIER ARELLANO ALAMEDA
Correo Electrónico	javier@issi.uned.es
Teléfono	91398-8735
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS

HORARIO DE ATENCIÓN AL ESTUDIANTE

1. Profesores Tutores (en el Grupo Intercampus correspondiente)

   Los tutorías son telemáticas, con la modalidad Intercampus. Son atendidas, en el Curso Virtual, por los Profesores Tutores Intercampus de la zona (Grupo de Tutoría) a la que el estudiante esté adscrito.

En algunos Centros Asociados también hay tutorías presenciales.

1. Equipo docente (en la Sede Central)

   Las guardias tendrán lugar todos los lunes y viernes **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 lectivos de 16h a 20h. Teléfono 913987792

   Javier Arellano Alameda

   Dpto. de Ingeniería de Software y Sistemas Informáticos. Despacho 2.21.
   Viernes lectivos de 16h a 20h. Teléfono 913988735

   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 Tutores es a través de los foros temáticos, que correspondan, en el Curso Virtual e, igualmente, con los profesores del equipo docente. En este caso, además, se puede utilizar 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.

TUTORIZACIÓN EN CENTROS ASOCIADOS

En el enlace que aparece a continuación se muestran los centros asociados y extensiones en las que se imparten tutorías de la asignatura. Estas pueden ser:

- **Tutorías de centro o presenciales**: se puede asistir físicamente en un aula o despacho del centro asociado.
- **Tutorías campus/intercampus**: se puede acceder vía internet.

Consultar horarios de tutorización de la asignatura 71013035

COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

Con la base de la *Memoria de Verificación del Título* y coherentemente con la exposición del epígrafe de '*Resultados de aprendizaje*', la asignatura incide en las siguientes competencias:

- Generales:
- G.1 - Competencias de gestión y planificación: Iniciativa y motivación. Planificación y organización (establecimiento de objetivos y prioridades, secuenciación y organización del tiempo de realización, etc.). Manejo adecuado del tiempo.
- G.2 - Competencias cognitivas superiores: selección y manejo adecuado de conocimientos, recursos y estrategias cognitivas de nivel superior apropiados para el afrontamiento y resolución de diversos tipos de tareas/problemas con distinto nivel de complejidad y novedad: Análisis y Síntesis. Aplicación de los conocimientos a la práctica Resolución de problemas en entornos nuevos o poco conocidos. Pensamiento creativo. Razonamiento crítico. Toma de decisiones.
- G.3 - Competencias de gestión de la calidad y la innovación: Seguimiento, monitorización y evaluación del trabajo propio o de otros. Aplicación de medidas de mejora. Innovación.

•G.6 - Trabajo en equipo. Trabajo en equipo desarrollando distinto tipo de funciones o roles. En la Sociedad del Conocimiento se presta especial atención a las potencialidades del trabajo en equipo y a la construcción conjunta de conocimiento, por lo que las competencias relacionadas con el trabajo colaborativo son particularmente relevantes: 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 (cuando se estime oportuno).

•Específicas:

•BC.1 - Capacidad para diseñar, desarrollar, seleccionar y evaluar, aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a los principios éticos y a la legislación y normativa vigente.

•BC.16 - Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

•BTEisw.1 - 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.

•BTEisw.4 - 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.

•BC.6 - Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.

•BC.7 - Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.

•BTEti.2 - Capacidad para seleccionar, diseñar, implantar, integrar, evaluar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.

•BC.3 - 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.

Las competencias anteriores son complejas. En algunos casos, la asignatura matiza y complementa aspectos que ya se han desarrollado parcialmente en las asignaturas mencionadas en el apartado de '*Contextualización*'. 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.

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:

- RA2. Conoce la Ingeniería de Requisitos y las técnicas específicas de diseño de software enfocadas al desarrollo orientado a objetos.
- RA4. 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.
- RA5. 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.
- RA6. Representa un diseño en relación a los requisitos cualitativos y las restricciones que debe cumplir.
- RA7. Analiza, valora y razona las ventajas e inconvenientes de un diseño utilizando criterios técnicos, cualitativos y del dominio del negocio.
- RA8. Es capaz de debatir y defender las conclusiones con argumentos sólidos y bien fundados.
- RA9. 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.
- RA10. 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 para el diseño OO de aplicaciones sencillas y para la asignación de responsabilidades a esos objetos. 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.
•Comprensión de los requisitos funcionales. Identificar las clases conceptuales, asociaciones y sus atributos en el dominio del problema y recogerlas en un modelo conceptual mediante la notación UML (Diseño de la Lógica del Funcionamiento deseado).
•Identificar eventos del sistema y representarlos en un diagrama de secuencia mediante UML.
•Diseño de la Dinámica del Funcionamiento Detallado. Asignar responsabilidades y diseñar colaboraciones entre los componentes del Software mediante los principios GRASP e ilustrar el resultado mediante diagramas de iteración en UML (secuencia detallado y colaboración).
•Resolver las definiciones de clases en un diagrama de clases en UML.
•Refinar el diseño y mejorar la arquitectura mediante la incorporación de parones 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.

CONTENIDOS

Unidad Didáctica I.

El hilo conductor del programa de esta asignatura es el libro de base "UML Y PATRONES. UNA INTRODUCCIÓN AL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS Y AL PROCESO UNIFICADO. SEGUNDA EDICIÓN". Sin embargo cualquier interpretación de sus contenidos no tiene porqué ajustarse a los objetivos del aprendizaje de la asignatura y, por tanto, a dicho programa.
Este libro presenta un proceso de desarrollo iterativo (UP) que organiza las actividades involucradas en el diseño, muestra cómo realizarlas, cómo representar las conclusiones y artefactos obtenidos en ellas, además de cómo obtener la especificación de un funcionamiento que satisfaga unos requisitos funcionales sencillos. En definitiva, muestra cómo hacer un diseño. Sin embargo, como ese diseño se podrían hacer otros muchos, y de otras maneras, por lo que, **lo más importante en relación con esta asignatura, sobre lo que el estudiante debe enfocar su atención, es que las indicaciones del libro están dirigidas a enseñar cómo conseguir que la solución funcional obtenida (el diseño) sea funcionalmente independiente, adaptable y comprensible**.

Tema 1. Introducción.

•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.

Tema 2. El Inicio.

•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 Diáctica II.

.

Tema 3. Primera iteración de la fase de elaboración.

•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 iteracción.
•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.

Tema 4. Segunda iteración de la fase de elaboración.

•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.

.

Tema 5. Tercera iteración de la fase de elaboración.

•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.

METODOLOGÍA

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. UNA INTRODUCCIÓN AL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS Y AL PROCESO UNIFICADO. SEGUNDA EDICIÓN").
•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 (SPdV)— 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.

•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.

•Acceso a la obtención del software de ayuda para modelado y representación gráfica Astah UML, concebido para usar UML.

•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 la prueba de evaluación continua.

•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—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 al alumnado.

•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 los foros de los Grupos de Tutorías a los que, respectivamente, estén adscritos.

SISTEMA DE EVALUACIÓN

TIPO DE PRUEBA PRESENCIAL

Tipo de examen	Examen de desarrollo
Preguntas desarrollo	10
Duración del examen	120 (minutos)

Material permitido en el examen

Original del libro de la asignatura "UML Y PATRONES. UNA INTRODUCCIÓN AL ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS Y AL PROCESO UNIFICADO. SEGUNDA EDICIÓN", de Craig Larman.

Criterios de evaluación

Las competencias de esta asignatura se refieren al diseño y, en la evaluación, se restringen al diseño detallado. En coherencia, **la solución que se pide en la evaluación consiste en una especificación de diseño detallado, formalmente correcta y cualitativamente adecuada**.

"Formalmente correcta" significa:

Que la especificación sea **completa**. Es decir, si no describe, completa y unívocamente, el código cuyo comportamiento representa (si no hay una representación adecuada del funcionamiento, que se pueda valorar), ni siquiera se ha aportado una solución al problema planteado.

Aunque la especificación esté completa, si el funcionamiento que describe no satisface las necesidades planteadas (es decir, si no funciona como debería, o no **resuelve el problema planteado**), el diseño tampoco es correcto.

"Cualitativamente adecuado" significa que el diseño, y el funcionamiento que especifica, sea:

Comprensible.

Adaptable (flexible).

Funcionalmente independiente: los elementos de código, en cualquier nivel de granularidad, deben tener una cohesión alta y, sobre todo, un **acoplamiento débil** entre ellos.

La prueba presencial (examen) tiene dos partes:

Las preguntas de 1ª a 8ª, con diferentes puntuaciones parciales y agrupadas en varias secciones, constituyen **NEX**. Su puntuación global es de 0 a 10. La prueba consiste en el desarrollo completo de un caso de uso; sencillo y, en la medida de lo posible, aislado de las interacciones con otros casos de uso. En las distintas preguntas (1ª a 7ª) **se pide** la ubicación del caso de uso en relación al sistema de estudio, su análisis y su diseño; de manera que, al final, **la especificación obtenida en el diseño lleve a un código** (en la 8ª pregunta) **cuyo comportamiento sea**, como mínimo, **el descrito en el enunciado para ese caso de uso**. Por tanto, la mayoría de las respuestas consisten en modelos y diagramas coherentes con la notación de UML. Es obvio que puede haber varias soluciones de diseño para el caso de uso planteado, pero se pide llegar a una definición, en su especificación, tal que no haya dudas sobre su validez.

Las preguntas 9ª y 10ª son **BP**, cuya puntuación global es de 0 a 1, y constituye un valor añadido que contribuye a la nota final en determinadas circunstancias (es decir, cuando **NEX** >= 6'25 y **PEC** aprobada).

% del examen sobre la nota final	100
Nota del examen para aprobar sin PEC	5
Nota máxima que aporta el examen a la calificación final sin PEC	9
Nota mínima en el examen para sumar la PEC	4,5
Comentarios y observaciones	

La contribución en la calificación final de la asignatura se diferencia para cada parte del examen:
La calificación de las preguntas 9ª y 10ª (**BP**) es una bonificación adicional y directa (entre 0 y 1 punto) en la nota final de la asignatura (**NF**, que siempre será <= 10) sólo para el caso de que **PEC** esté aprobada **y** **NEX** >= 6'25. En otros casos no se considera.
La contribución de la suma de las calificaciones obtenidas en las preguntas 1ª a 8ª (**NEX**) a **NF** depende del propio valor de **NEX**; según un algoritmo por tramos con el que se pretende que la calificación obtenida en la **PEC**, cuando esté aprobada, siempre beneficie al estudiante elevando su nota final:

Si **PEC** <5, **NF** = **NEX**
Si **PEC** >= 5:
Si **NEX** <4'5, **NF** = **NEX**
Si 4'5 <= **NEX** <5, **NF** = **NEX** + 0'2 * **PEC**
Si 5 <= **NEX** <6'25, **NF** = máx { (**NEX**) | (0'8***NEX** + 0'2***PEC**) }
Si **NEX** >= 6'25, **NF** = máx { (**NEX**) | (0'8***NEX** + 0'2***PEC** + **BP**) }

PRUEBAS DE EVALUACIÓN CONTINUA (PEC)

¿Hay PEC? Si

Descripción

La Prueba de Evaluación Continua es única en cada curso y consiste en el desarrollo completo de un caso de uso; sencillo y, en la medida de lo posible, aislado de las interacciones con otros casos de uso. Tanto en su planteamiento como en las preguntas que articulan las distintas actividades que contiene, la prueba es idéntica a las preguntas 1ª a 8ª de la Prueba Presencial (examen).

No es una prueba obligatoria, ni excluye la posibilidad de aprobar la asignatura.

Criterios de evaluación

Los objetivos que se evalúan son idénticos a los del examen. Es decir, en las distintas preguntas (1ª a 7ª) **se pide** la ubicación del caso de uso en relación al sistema de estudio, su análisis y su diseño; de manera que, al final, la especificación obtenida en el diseño lleve a un código (en la 8ª pregunta) cuyo comportamiento sea, como mínimo, el descrito en el enunciado para ese caso de uso.

Son 8 preguntas, con diferentes puntuaciones parciales y agrupadas en varias secciones, cuya valoración global (de la prueba) estará entre 0 y 10 (PEC).
La prueba está superada si la calificación obtenida es PEC >= 5.

Ponderación de la PEC en la nota final	Sólo en el caso de la prueba esté aprobada, contribuirá a la nota final con el 20% de la calificación obtenida; tanto en la convocatoria ordinaria de febrero como, si se da el caso, en la extraordinaria de septiembre.
Fecha aproximada de entrega	10/01/2021

Comentarios y observaciones

El resultado del trabajo, con las respuestas de la prueba, se redactará en un único documento con el formato de '*memoria*' (en el curso virtual se encontrará una plantilla para dicho formato) y se entregará en el área de '*Entrega de trabajos*' del mencionado curso virtual, **con anterioridad a la fecha límite fijada** para ello.

Aunque el planteamiento de la prueba, sus preguntas y los criterios de evaluación son idénticos a los del examen, las condiciones en las que se elaboran las repuestas son muy diferentes y, al contrario que el examen, esta prueba es revisada por el Tutor asignado al estudiante. Por ello, la calificación obtenida en la PEC en ningún caso puede servir de referencia para extrapolar la nota '*que se debería*' alcanzar en el examen. Por los mismos motivos, tampoco se admite evaluar los trabajos que se entreguen después de la fecha límite (anterior al período de exámenes de la convocatoria de febrero) ni cualesquiera otros que se pretenda sean su sustituto. Sin embargo, se utilizará la misma calificación obtenida en el trabajo, depositado correctamente en tiempo y forma, para el cálculo de la nota final de la asignatura, tanto en la convocatoria ordinaria de febrero como, si se da el caso, en la extraordinaria de septiembre.

En cuanto a la contribución de la calificación obtenida en el cómputo de la nota final de la asignatura, será del 20% sólo en el caso de que esté aprobada (PEC >= 5) y la calificación obtenida en la primera parte del examen (NEX, preguntas 1^{a} a 8^{a}) sea NEX >= 4'5.

OTRAS ACTIVIDADES EVALUABLES

¿Hay otra/s actividad/es evaluable/s?	No
Descripción	
Criterios de evaluación	
Ponderación en la nota final	0
Fecha aproximada de entrega	
Comentarios y observaciones	

¿CÓMO SE OBTIENE LA NOTA FINAL?

La nota final de la asignatura (**NF**) estará comprendida entre 0 y 10; y la asignatura está superada cuando **NF** >= 5.

PEC es la calificación obtenida en la prueba de evaluación continua. Está comprendida entre 0 y 10. No es obligatoria ni excluye la posibilidad de aprobar la asignatura. Sólo se evaluará y se tendrá en cuenta si la entrega se realiza dentro de los plazos fijados. De igual forma, contribuye a **NF** cuando **PEC** >= 5 **y** **NEX** >= 4'5 y, en esos casos, lo hace con el 20% de su valor.

BP es el valor obtenido como suma de las calificaciones de las respuestas a las preguntas 9^a y 10^a del examen presencial. Está comprendido entre 0 y 1. Sólo contribuye a **NF** cuando **PEC** >= 5 **y** **NEX** >= 6'25 y, en esos casos, lo hace con el 100% de su valor (siempre que **NF** no resulte mayor que 10).

NEX es el valor obtenido como suma de las calificaciones de las respuestas a las preguntas 1^a a 8^a del examen presencial. Está comprendido entre 0 y 10. Cuando **NEX** <5, su contribución a **NF** es del 100%. A partir de ese valor (5), su contribución es del 80% o del 100% dependiendo de la situación más favorable para el estudiante en la que se pueda compaginar, o no, con las contribuciones de **PEC** y **BP**.

Para resumir e ilustrar la aplicación del algoritmo, se incluye este esquema. En cualquier caso, una vez fijadas las condiciones que delimitan los tramos y la participación de los componentes de la nota, la aplicación del algoritmo se realiza para que el estudiante obtenga el mejor resultado posible.

Si **PEC** <5, **NF** = **NEX**

Si **PEC** >= 5:

Si **NEX** <4'5, **NF** = **NEX**

Si 4'5 <= **NEX** <5, **NF** = **NEX** + 0'2 * **PEC**

Si 5 <= **NEX** <6'25, **NF** = máx { (**NEX**) | (0'8***NEX** + 0'2***PEC**) }

Si **NEX** >= 6'25, **NF** = máx { (**NEX**) | (0'8***NEX** + 0'2***PEC** + **BP**) }

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

Nota importante: Este libro presenta un proceso de desarrollo iterativo (UP) que organiza las actividades involucradas en el diseño, muestra cómo realizarlas, cómo representar las conclusiones y artefactos obtenidos en ellas, además de cómo obtener la especificación de un funcionamiento que satisfaga unos requisitos funcionales sencillos. En definitiva, muestra cómo hacer un diseño. Sin embargo, como ese diseño se podrían hacer otros muchos, y de otras maneras, por lo que, **lo más importante en relación con esta asignatura, sobre lo**

que el estudiante debe enfocar su atención, es que las indicaciones del libro están dirigidas a enseñar cómo conseguir que la solución funcional obtenida (el diseño) sea funcionalmente independiente, adaptable y comprensible.

BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13):9788177589795
Título:APPLYING UML AND PATTERNS: AN INTRODUCTION TO OBJECT-ORIENTED ANALYSIS AND DESIGN AND ITERATIVE DEVELOPMENT (3)
Autor/es:Larman, C. ;
Editorial:PEARSON EDUCATION

ISBN(13):9788478290598
Título:PATRONES DE DISEÑO
Autor/es:Gamma, Erich ; Otros ;
Editorial:PEARSON ADDISON-WESLEY

ISBN(13):9789684443648
Título:UML GOTA A GOTA
Autor/es:Fowler, Martin ; Kendall, Scott ;
Editorial:PEARSON

RECURSOS DE APOYO Y WEBGRAFÍA

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

- El curso virtual, que constituye el principal punto de apoyo. Contiene múltiples recursos de comunicación y aprendizaje. Además de los ya mencionados en otros apartados, en lo que se refiere a los recursos de apoyo adicionales, contiene:
- Un servicio de repositorio documental, para disponer de materiales y documentación adicionales. Entre ellos, la relación y descripción de las pruebas de evaluación continua (PEC) de cursos anteriores, así como las guías y propuestas de solución publicadas para ellas.
- Dentro el plazo de elaboración, el enunciado de la PEC, plantillas, guías, normas, criterios de corrección y condiciones que se deben tener en cuenta para la elaboración y entrega de dicha actividad. Tras el mencionado plazo, además, las soluciones propuestas para la PEC, junto con recomendaciones para la corrección de las desviaciones observadas, con más frecuencia, en la comprensión de la asignatura.
- Las tutorías, en la modalidad Intercampus, en las que el profesor tutor resuelve dudas y cuestiones académicas y explica aquellos conceptos y/o temas que resultan más

complicados. Dentro de esta modalidad, y bajo suscripción al curso virtual de la asignatura, también se dispone de una colección de tutorías en Webconferencia; almacenadas en el repositorio de contenidos audiovisuales (GICA) de INTECCA (https://www.intecca.uned.es/portalavip/grabaciones.php"71013035 - DISEÑO DEL SOFTWARE - Intercampus ...", requiere suscripción).

- En la página Web de la asignatura —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 UML (requiere registro para obtener licencia anual de estudiante).
- 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.

IGUALDAD DE GÉNERO

En coherencia con el valor asumido de la igualdad de género, todas las denominaciones que en esta Guía hacen referencia a órganos de gobierno unipersonales, de representación, o miembros de la comunidad universitaria y se efectúan en género masculino, cuando no se hayan sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.