



## 17832 - ANÁLISIS Y DISEÑO DE SOFTWARE

### Información de la asignatura

**Código - Nombre:** 17832 - ANÁLISIS Y DISEÑO DE SOFTWARE

**Titulación:** 473 - Graduado/a en Ingeniería Informática  
474 - Graduado/a en Ingeniería Informática y Matemáticas  
722 - Graduado/a en Ingeniería Informática (Modalidad Bilingüe 2018)  
734 - Graduado/a en Ingeniería Informática y Matemáticas (2019)

**Centro:** 350 - Escuela Politécnica Superior

**Curso Académico:** 2020/21

### 1. Detalles de la asignatura

#### 1.1. Materia

Ingeniería del software

#### 1.2. Carácter

Obligatoria

#### 1.3. Nivel

Grado (MECES 2)

#### 1.4. Curso

2

#### 1.5. Semestre

Segundo semestre

#### 1.6. Número de créditos ECTS

6.0

#### 1.7. Idioma

Español

#### 1.9. Recomendaciones

Esta asignatura asume conocimientos de programación estructurada por parte del estudiante, que pueden haber sido adquiridos cursando las materias del módulo de Programación y Estructuras de Datos del plan de estudios. Es muy aconsejable haber superado con éxito las asignaturas Programación I, Programación II y Proyecto de Programación para un aprovechamiento de la asignatura ADS. La materia Análisis y Diseño de Software se desglosa en dos asignaturas: Análisis y

Código Seguro de Verificación:		Fecha:	11/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	1/9	

Diseño de Software y Proyecto de Análisis y Diseño de Software, ambas impartidas en el 2º cuatrimestre del segundo curso. Ya que ambas proporcionan formación complementaria y conceptos relacionados, se recomienda cursarlas a la vez.

Se recomienda para garantizar la asimilación de los contenidos y la adquisición de habilidades la lectura crítica de los textos de la bibliografía, el uso del material electrónico de esta asignatura disponible en la plataforma Moodle y la búsqueda activa de material complementario en la red. Es recomendable disponer de un dominio de inglés que permita al alumno leer la bibliografía de consulta. Asimismo, se requiere iniciativa personal y constancia para el diseño e implementación de aplicaciones, así como predisposición y empatía para el trabajo colaborativo en grupo.

### 1.10. Requisitos mínimos de asistencia

Se plantean dos itinerarios, uno con asistencia obligatoria a clase y otro sin ella, los estudiantes deberán optar por uno u otro dentro de las primeras dos semanas y cumplir con los distintos requisitos de evaluación que conlleva cada uno de los modelos, publicados en la presente guía docente. Si un estudiante desea pasar de itinerario continuo a final, después de las dos primeras semanas, podrá hacerlo.

#### ITINERARIO CON ASISTENCIA OBLIGATORIA A CLASE

La asistencia es obligatoria al menos en un 85%.

#### ITINERARIO SIN ASISTENCIA OBLIGATORIA A CLASE

La asistencia es muy recomendable aunque no obligatoria.

### 1.11. Coordinador/a de la asignatura

Juan de Lara Jaramillo

<https://autoservicio.uam.es/paginas-blancas/>

### 1.12. Competencias y resultados del aprendizaje

#### 1.12.1. Competencias

**C1** Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

**C2** Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.

**C3** Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software.

**C4** Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.

**C5** Conocimiento, administración y mantenimiento sistemas, servicios y aplicaciones informáticas.

**C8** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

**C16** Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

**C17** Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

**IS1** 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.

#### 1.12.2. Resultados de aprendizaje

Analizar, Diseñar, Implementar y Probar programas usando tecnologías de Orientación a Objetos.

Conocer y aplicar buenas prácticas de Ingeniería del Software en el desarrollo de aplicaciones.

#### 1.12.3. Objetivos de la asignatura

Los objetivos específicos que se pretenden alcanzar con esta asignatura, en los distintos temas, son:

TEMA 1.- Ciclo de Vida del Software

Código Seguro de Verificación:		Fecha:	11/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	2/9	

1.	Comprender la utilidad de las distintas fases de un proyecto: desarrollar software no es sólo programar.
2.	Comparar las distintas metodologías y modelos de desarrollo de software, y ser capaz de seleccionar los más adecuados a cada tipo de proyecto.
TEMA 2.- Orientación a Objetos	
1.	Conocer los conceptos fundamentales del diseño orientado a objetos.
2.	Comparar el enfoque procedimental con el orientado a objetos, y entender las ventajas y limitaciones de cada uno.
3.	Aplicar notaciones de diseño (UML) para reflejar la estructura y comportamiento de un sistema software orientado a objetos.
TEMA 3.- Diseño e Implementación en Java	
1.	Comparar el lenguaje Java con C, entender su modo de funcionamiento, sus ventajas y limitaciones.
2.	Diseñar una aplicación a nivel básico: clases y objetos, y su implementación en Java.
3.	Diseñar la jerarquía de herencia, así como dominar y aplicar el concepto de interfaz en el diseño orientado a objetos.
4.	Diseñar el control de acceso y la organización modular de una aplicación.
5.	Aprender a usar las Interfaces
6.	Diseño e implementación de la gestión de errores, y el manejo de excepciones.
7.	Aprender a usar las distintas Colecciones de la librería y los parámetros Genéricos
8.	Comprender como funcionan las Clases Internas y la Reflexión.

<b>Código Seguro de Verificación:</b>		<b>Fecha:</b>	11/01/2021	
<b>Firmado por:</b>	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
<b>Url de Verificación:</b>		<b>Página:</b>	3/9	

9.	Uso de expresiones Lambda e interfaces funcionales
TEMA 4.- Patrones de Diseño	
1.	Comprender el papel fundamental de los patrones de diseño y la reutilización en la construcción de software.
2.	Utilizar patrones de diseño en el desarrollo de aplicaciones.

### 1.13. Contenidos del programa

#### Programa Sintético

UNIDAD 1. Ciclo de Vida del Software

UNIDAD 2. Orientación a Objetos

UNIDAD 3. Diseño e Implementación en Java

UNIDAD 4. Patrones de Diseño

#### Programa Detallado

1. Ciclo de Vida del Software
  1. Fases en el desarrollo del Software
  2. Modelos de Ciclo de Vida
  3. Metodologías
2. Orientación a Objetos
  1. Conceptos de Diseño Orientado a Objetos
  2. Objetos y Clases
  3. Encapsulamiento
  4. Herencia y Polimorfismo
  5. Notaciones de Diseño.
3. Diseño e Implementación en Java
  1. Introducción a Java
  2. Clases y Objetos en Java
  3. Jerarquías de clases.
    1. Subclases y Herencia
    2. Polimorfismo.
    3. Métodos abstractos.
    4. Clases anónimas.
  4. Control de acceso, modularidad.
  5. Interfaces.
  6. Entrada/Salida. Tratamiento de errores. Excepciones.
  7. Colecciones y Genericidad.
  8. Clases Internas y Reflexión.
  9. Expresiones Lambda e interfaces funcionales.
4. Patrones de Diseño
  1. Introducción.
  2. Patrones de Creación.

Código Seguro de Verificación:		Fecha:	11/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	4/9	

3. Patrones Estructurales.
4. Patrones de Comportamiento.

## 1.14. Referencias de consulta

Nota: Esta asignatura no sigue ningún libro en concreto. La lectura recomendada se lista por orden de afinidad al contenido del programa.

Unidad 1:

1. [Software engineering a practitioner's approach](#), 7ªed. Roger Pressman, McGraw Hill Higher Education, 2010. INF/C6110B/PRE. También disponible en castellano. Capítulos 1 y 2.
2. [Software engineering](#), 9ª ed. Ian Sommerville, Addison Wesley, 2011. INF/C6110B/SOM. También disponible en castellano. Capítulos 1 y 2.

Unidad 2:

3. [Ingeniería de software clásica y orientada a objetos](#), Sexta Edición. Stephen Schach. McGraw-Hill. INF/C6110B/SCH.
4. [Construcción de software orientado a objetos](#). Betrand Meyer. Prentice Hall. INF/C6110J/MEY.
5. [El lenguaje unificado de modelado manual de referencia](#). Rumbaugh, James. Pearson Addison Wesley. 2007. INF/C6140-U/RUM.

Unidad 3:

6. Core Java Vol. I Fundamentals, Horstmann, Cay S. Prentice Hall, 2015. INF/C6140-J/HOR Vol. 1. También disponible en castellano.
7. [Core Java Vol. II Advanced features](#), Horstmann, Cay S. Prentice Hall, 2015. INF/C6140-J/HOR Vol. 2. También disponible en castellano.
8. Functional programming in Java harnessing the power of Java 8 Lambda expressions, Subramaniam, Venkat. Disponible electrónicamente en la biblioteca de la EPS.

Unidad 4:

9. [Patrones de diseño elementos de software orientado a objetos reutilizable](#). Gamma, E., Helm, R., Johnson, R., Vlissides, J. INF/C6110J/PAT. Addison-Wesley, 2003.
10. [Patrones de diseño aplicados a Java](#). Stelling, Stephen. INF/C6140-J/STE. Pearson, Prentice Hall. 2003.
11. [UML y patrones una introducción al análisis y diseño orientado a objetos y al proceso unificado](#) (2ª edición). Craig Larman. Prentice Hall, 2002. INF/C6140-U/LAR. También disponible en formato electrónico.

Nota: no se recomienda a los estudiantes comprar ningún libro hasta no haber comparado su contenido con el programa y revisado previamente en la biblioteca.

Material electrónico de trabajo: los documentos electrónicos de trabajo (recomendaciones sobre la elaboración de documentación, realización de diagramas, recomendaciones sobre legibilidad en el código, ejercicios del curso, etc.) se publican en la sección de ADS en plataforma Moodle (<http://moodle.uam.es>)

## 2. Metodologías docentes y tiempo de trabajo del estudiante

### 2.1. Presencialidad

	#horas
Porcentaje de actividades presenciales (mínimo 33% del total)	75
Porcentaje de actividades no presenciales	75

### 2.2. Relación de actividades formativas

Actividades presenciales	Nº horas
Clases teóricas en aula	42
Seminarios	
Clases prácticas en aula	
Prácticas clínicas	

Código Seguro de Verificación:		Fecha:	11/01/2021	
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva			
Url de Verificación:		Página:	5/9	

Prácticas con medios informáticos	26
Prácticas de campo	
Prácticas de laboratorio	
Prácticas externas y/o practicum	
Trabajos académicamente dirigidos	
Tutorías	5
Actividades de evaluación	2
Otras	

### 3. Sistemas de evaluación y porcentaje en la calificación final

#### 3.1. Convocatoria ordinaria

- Ambas partes, teoría y prácticas se puntúan sobre 10 puntos.
- La nota final de la asignatura se obtiene de las notas de teoría y prácticas por medio de la ecuación:

Calificación:  $0.3 \cdot \text{Prácticas} + 0.7 \cdot \text{Teoría}$

- Para aprobar la asignatura es obligatorio obtener una nota mayor o igual a 5 puntos, tanto en la parte de teoría como en las prácticas. En caso contrario, la nota final en actas será

Calificación:  $(0.3 \cdot \text{Mín}(5, \text{Prácticas}) + 0.7 \cdot \text{Mín}(5, \text{Teoría}))$

La nota correspondiente a la parte de Teoría es la que resulta del máximo de:

- La calificación de la prueba final (60%) y la calificación de las pruebas/actividades/ejercicios intermedios (40%).
- La calificación de la prueba final.

La calificación de las pruebas intermedias se calculará mediante la media ponderada de cada prueba, con una ponderación comprendida entre el 30% y el 70%. Dicha ponderación se publicará junto con el calendario de realización de las mismas.

Para ser evaluado por la modalidad continua, es necesario tener una media de 4.0 o más en las pruebas intermedias, y haberse presentado a todas ellas.

La nota correspondiente a la parte de Teoría para el itinerario sin asistencia obligatoria corresponde únicamente a la prueba final (distinta a la de la trayectoria con asistencia obligatoria ya que abarca ejercicios relacionados con las actividades intermedias).

Las pruebas escritas, podrán incluir tanto cuestiones teóricas y ejercicios como la realización de diagramas, y el diseño y escritura de programas.

- La nota correspondiente a la parte de prácticas es la que resulta de realizar las prácticas programadas en el curso.
- Para aprobar la parte práctica el estudiante deberá asistir, al menos, al 85% de las prácticas. En caso contrario deberá realizar un examen de prácticas consistente en una práctica de mayor complejidad a las realizadas en el laboratorio.
- Para aprobar la parte práctica, es imprescindible entregar cada una de las prácticas y obtener al menos una nota de 3,5 puntos en cada una. En caso de que esto no se cumpla, por faltar alguna entrega, no llegar a la nota mínima en cada una, o no hacer el examen de prácticas en el caso de evaluación no continua, la nota de prácticas será el  $\text{Min}(4, \text{media ponderada de las prácticas})$ .
- La calificación de la parte práctica tendrá en cuenta la calidad de los diseños realizados y el nivel de los resultados obtenidos. También se valorará la validez de los resultados obtenidos en cada uno de los apartados que se hayan establecido para su realización en los guiones de las prácticas.
- Las notas tanto de teoría como de prácticas se conservan sólo para la convocatoria extraordinaria del mismo curso académico.
- El número mínimo de pruebas a las que el estudiante se ha de presentar para recibir una calificación numérica es 1. Por debajo de este número el estudiante recibirá la calificación "No evaluado". Aunque el estudiante no se presente a la prueba final, siempre que se haya presentado a este mínimo de pruebas, recibirá una calificación numérica.

**ATENCIÓN:** Cualquier copia descubierta que se haya realizado a lo largo del curso, tanto en cualquiera de las actividades de teoría desarrolladas, como en cualquiera de los apartados de las prácticas, serán penalizadas con rigurosidad. La normativa de la UAM establece que cuando un profesor observe conductas o actos en un estudiante incompatibles con la probidad y la ética, con independencia de su posible repercusión en la calificación de la prueba, podrá solicitar del Rector la incoación del correspondiente expediente informativo al estudiante en cuestión.

##### 3.1.1. Relación actividades de evaluación

Actividad de evaluación	%
-------------------------	---

Código Seguro de Verificación:		Fecha:	11/01/2021
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva		
Url de Verificación:		Página:	6/9

Examen final (máximo 70% de la calificación final o el porcentaje que figure en la memoria)	60%
Evaluación continua	40%

### 3.2. Convocatoria extraordinaria

- Ambas partes, teoría y prácticas se puntúan sobre 10 puntos.
- La nota final de la asignatura se obtiene de las notas de teoría y prácticas por medio de la ecuación:

Calificación:  $0.3 \cdot \text{Prácticas} + 0.7 \cdot \text{Teoría}$

- Para aprobar la asignatura es obligatorio obtener una nota mayor o igual a 5 puntos, tanto en la parte de teoría como en las prácticas. En caso contrario, la nota final en actas será

Calificación:  $(0.3 \cdot \text{Mín}(5, \text{Prácticas}) + 0.7 \cdot \text{Mín}(5, \text{Teoría}))$

Las pruebas escritas, podrán incluir tanto cuestiones teóricas y ejercicios como la realización de diagramas, y el diseño y escritura de programas.

- El número mínimo de pruebas a las que el estudiante se ha de presentar para recibir una calificación numérica es 1. Por debajo de este número el estudiante recibirá la calificación "No evaluado". Aunque el estudiante no se presente a la prueba final, siempre que se haya presentado a este mínimo de pruebas, recibirá una calificación numérica.

**ATENCIÓN:** Cualquier copia descubierta que se haya realizado a lo largo del curso, tanto en cualquiera de las actividades de teoría desarrolladas, como en cualquiera de los apartados de las prácticas, serán penalizadas con rigurosidad. La normativa de la UAM establece que cuando un profesor observe conductas o actos en un estudiante incompatibles con la probidad y la ética, con independencia de su posible repercusión en la calificación de la prueba, podrá solicitar del Rector la incoación del correspondiente expediente informativo al estudiante en cuestión.

#### 3.2.1. Relación actividades de evaluación

Examen 100%

### 4. Cronograma orientativo

Semana	Contenido	Horas presenciales	Horas no presenciales (Trabajo autónomo del estudiante)
1	- Presentación de la asignatura, descripción del programa, normativa y los métodos de evaluación.  - Unidad 1. Ciclo de Vida del Software. Temas 1.1, 1.2 y 1.3	3	5  Lectura de las normativas de teoría y prácticas.  Estudio del material de la Unidad 1.
2	- Unidad 2 Orientación a objetos. Temas 2.1, 2.2 y 2.3  - Práctica 1: Introducción a Java.	5	3  Estudio del material de la Unidad 2.  Realización de los ejercicios propuestos.  Realización de Práctica 1.
3	- Unidad 2 Orientación a Objetos. Temas 2.4 y 2.5  - Práctica 2: Introducción al diseño orientado a objetos.	4	2  Estudio del material de la Unidad 2.  Realización de los ejercicios propuestos.  Entrega de la Práctica 1.  Realización de Práctica 2.
3	Prueba #1: Análisis y conceptos de orientación a	1	

<b>Código Seguro de Verificación:</b>		<b>Fecha:</b>	11/01/2021
<b>Firmado por:</b>	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva		
<b>Url de Verificación:</b>		<b>Página:</b>	7/9

	objetos. Tutoría sobre las Unidades 1 y 2.	1	
4	- Unidad 3. Diseño e Implementación en Java. Tema 3.1. - Práctica 2: Introducción al diseño orientado a objetos.	5	3 Estudio del material de la Unidad 3. Realización de los ejercicios. Realización de Práctica 2.
5	- Unidad 3. Diseño e Implementación en Java. Tema 3.2. -Práctica 3: Clases y Objetos Java.	5	3 Estudio del material de la Unidad 3. Realización de los ejercicios. Entrega de la Práctica 2. Realización de Práctica 3.
6	-Unidad 3. Diseño e Implementación en Java. Tema 3.3. -Práctica 3: Clases y Objetos Java.	5	3 Estudio del material de la Unidad 3. Realización de los ejercicios. Realización de Práctica 3.
7	-Unidad 3. Diseño e Implementación en Java. Tema 3.4. -Práctica 3: Clases y Objetos Java.	5	3 Estudio del material de la Unidad 3. Realización de los ejercicios. Realización de Práctica 3.
8	-Unidad 3. Diseño e Implementación en Java. Tema 3.5 -Práctica 4: Herencia, interfaces y excepciones.	4	2 Estudio del material de la Unidad 3. Realización de los ejercicios. Entrega Práctica 3. Realización Práctica 4.
9	-Unidad 3. Diseño e Implementación en Java. Tema 3.6. -Práctica 4: Herencia, interfaces y excepciones	5	3 Estudio del material de la Unidad 3. Realización de los ejercicios. Realización Práctica 4.
9	Prueba #2: Java Tutoría.	1 1	
10	-Unidad 3. Diseño e Implementación en Java. Tema 3.7. -Práctica 4: Herencia, interfaces y excepciones.	5	3 Estudio del material de la Unidad 3. Realización de los ejercicios. Realización de Práctica 4.
11	-Unidad 3. Diseño e Implementación en Java. Tema 3.8. -Práctica 5: Genericidad, expresiones lambda y patrones de diseño.	4	2 Estudio del material de la Unidad 3. Realización de los ejercicios. Entrega Práctica 4.

<b>Código Seguro de Verificación:</b>		<b>Fecha:</b>	11/01/2021	
<b>Firmado por:</b>	<i>Esta guía docente no está firmada mediante CSV porque no es la versión definitiva</i>			
<b>Url de Verificación:</b>		<b>Página:</b>	8/9	



			Realización de Práctica 5.
11	Tutoría	1	
12	-Unidad 3. Diseño e Implementación en Java. Tema 3.9. -Práctica 5: Genericidad, expresiones lambda y patrones de diseño.	5	3 Estudio del material de la Unidad 3. Realización de los ejercicios. Realización de Práctica 5.
13	-Unidad 4. Patrones de Diseño -Práctica 5: Genericidad, expresiones lambda y patrones de diseño.	5	3 Estudio del material de la Unidad 4. Realización de los ejercicios. Realización y entrega de Práctica 5.
14	- Tutorías sobre los temas 3 y 4.	1	
14	-Unidad 4. Patrones de Diseño - Repaso y preparación del examen	5	2 Estudio del material de la Unidad 4. Realización de los ejercicios.
	Examen Final	3	16h

Código Seguro de Verificación:		Fecha:	11/01/2021
Firmado por:	Esta guía docente no está firmada mediante CSV porque no es la versión definitiva		
Url de Verificación:		Página:	9/9