

23-24

GRADO EN INGENIERÍA EN  
TECNOLOGÍAS DE LA INFORMACIÓN  
SEGUNDO CURSO

# GUÍA DE ESTUDIO PÚBLICA



## DISEÑO DE APLICACIONES ORIENTADAS A OBJETOS

CÓDIGO 71022011

UNED

23-24

DISEÑO DE APLICACIONES ORIENTADAS A  
OBJETOS

CÓDIGO 71022011

# 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 DE APLICACIONES ORIENTADAS A OBJETOS
Código	71022011
Curso académico	2023/2024
Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS
Título en que se imparte	GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
CURSO - PERIODO	GRADO EN INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN - SEGUNDO CURSO - SEMESTRE 2
CURSO - PERIODO	ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN EN UNED - OPTATIVAS CURSO - SEMESTRE 2
CURSO - PERIODO	ESPECÍFICO PARA INGENIEROS TÉCNICOS EN INFORMÁTICA DE GESTIÓN - OPTATIVAS CURSO - SEMESTRE 2
Tipo	OBLIGATORIAS
Nº ETCS	6
Horas	150.0
Idiomas en que se imparte	CASTELLANO

## PRESENTACIÓN Y CONTEXTUALIZACIÓN

El objetivo de esta guía es orientar al alumno en el estudio de la asignatura. Se recomienda la lectura completa de la guía a comienzo del cuatrimestre para tener una idea completa de la temática de la asignatura y el calendario de prácticas, de forma que el alumno pueda planificar su trabajo para cumplir con las fechas de entrega.

La asignatura de Diseño de Aplicaciones Orientada a Objetos (DAOO en adelante) se imparte en el segundo cuatrimestre del segundo curso, consta de 6 créditos, con carácter de obligatoria, para la titulación de grado en Tecnologías de la Información.

Esta asignatura es una introducción al Diseño de Aplicaciones Orientadas a Objetos (DAOO) y tiene como objetivo principal el estudio y la puesta en práctica de los patrones de diseño como abstracción para la realización de aplicaciones y sistemas complejos basados en la metodología de orientación a objetos. Dado que el alumno ha tomado contacto con esta metodología en la asignatura de Programación Orientada a Objetos (POO), en este curso se profundizará en su uso y aplicación en escenarios más complejos. Así mismo, y dado que en POO el alumno ha desarrollado programas básicos utilizando el lenguaje de programación Java y la herramienta didáctica BlueJ, la asignatura DAOO mantendrá el uso de este lenguaje de programación motivando al alumno a implementar las actividades propuestas en el temario utilizando Eclipse como entorno de desarrollo estándar y muy extendido en la comunidad de programadores.

Gracias a esta asignatura, un alumno va a ver el valor de aplicar soluciones reutilizables de software a nuevos problemas. Los patrones presentados aquí son el resultado de años de desarrollo y perfeccionamiento y representan elementos de software robustos y aplicables en un amplio número de escenarios. Por lo tanto, la asignatura es muy relevante tanto para los alumnos en el estudio de otras asignaturas en su grado como para su carrera profesional.

## REQUISITOS Y/O RECOMENDACIONES PARA CURSAR LA ASIGNATURA

Al tratarse de una continuación de asignatura POO, se requiere que el alumno sabe lo básico del diseño y desarrollo usando la orientación a objetos y como hacer programas con Java.

## EQUIPO DOCENTE

Nombre y Apellidos	TIMOTHY MARTIN READ (Coordinador de asignatura)
Correo Electrónico	tread@lsi.uned.es
Teléfono	91398-8261
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS
Nombre y Apellidos	JUAN MANUEL CIGARRAN RECUERO
Correo Electrónico	juanci@lsi.uned.es
Teléfono	91398-9828
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS
Nombre y Apellidos	COVADONGA RODRIGO SAN JUAN
Correo Electrónico	covadonga@lsi.uned.es
Teléfono	91398-6487
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS

## HORARIO DE ATENCIÓN AL ESTUDIANTE

Horario de asistencia al estudiante:

- Juan Cigarrán: Jueves 16.00 - 20:00
- Timothy Read: Miércoles 09:30 - 13:30

Datos de contacto:

- Juan Cigarrán  
Dpto. de Lenguajes y Sistemas Informáticos  
ETSI Informática  
C./ Juan del Rosal, 16  
28040 Madrid  
Tlfno.: 91 398 7620  
Email: juanci@lsi.uned.es
- Timothy Read  
Dpto. de Lenguajes y Sistemas Informáticos  
ETSI Informática

C./ Juan del Rosal, 16  
28040 Madrid  
Tlfno.: 91 398 8261  
Email: tread@lsi.uned.es

Dirección postal:

Equipo docente de la asignatura Diseño de Aplicaciones Orientada a Objetos

Dpto. de Lenguajes y Sistemas Informáticos

ETSI Informática

C./ Juan del Rosal, 16

28040 Madrid

## 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 71022011

## COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

Con el estudio de la asignatura el alumno debe adquirir una visión general de la DAOO. En concreto, se pretende que al finalizar el curso el alumno: a) Haya reforzado y ampliado sus conocimientos de programación orientada a objetos, b) Conozca los patrones de uso general de creación, estructura y comportamiento, c) Sea capaz de identificarlos en la etapa de diseño de una aplicación software, y; d) Conozca el modo en el que estos se implementan en el lenguaje Java.

Competencias específicas a la asignatura

FB.5 - Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, así como de los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería

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.

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

BTEti.3 - Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas

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

## RESULTADOS DE APRENDIZAJE

Con el estudio de la asignatura el alumno debe adquirir una visión general de la DAOO. En concreto, se pretende que al finalizar el curso el alumno: a) Haya reforzado y ampliado sus conocimientos de programación orientada a objetos, b) Conozca los patrones de uso general de creación, estructura y comportamiento, c) Sea capaz de identificarlos en la etapa de diseño de una aplicación software, y; d) Conozca el modo en el que estos se implementan en el lenguaje Java.

Los resultados de aprendizaje que los alumnos van a conseguir en esta asignatura, según la memoria de verificación de ANECA son:

- R4. Realizar pruebas de validación y depuración de un programa dado.
- R5. Identificar clases de objetos con los datos de un problema.
- R6. Crear clases y objetos y manipularlos.
- R7. Identificar los métodos que caracterizan a una clase de objetos.
- R8. Utilizar el mecanismo de la herencia para crear clases.
- R9. Identificar las relaciones entre clases en distintos casos de uso.
- R10. Diseñar y documentar interfaces de usuario.
- R11. Usar Patrones de Diseño
- R12. Abordar metodológicamente proyectos de programación medios.
- R13. Dominar un lenguaje de programación acorde al paradigma del contenido de la materia.

## CONTENIDOS

Parte 1. El diseño orientado a objetos

Parte 2. Los patrones de diseño

## METODOLOGÍA

La enseñanza a distancia posee unas características que la diferencian de la presencial. Sin embargo, esto no impide que el alumno pueda disponer de la ayuda y los recursos necesarios para cursar las asignaturas en las que se matricule.

Los mecanismos que tiene el alumno para facilitar el aprendizaje requerido en la asignatura son los siguientes:

- Tutorías presenciales o virtuales en el centro asociado correspondiente. Se puede consultar los horarios de tutorización de la asignatura en el siguiente enlace:

[https://akademosweb.uned.es/Default.aspx?id\\_servicio=19&modo\\_23=1&asigna=71022011](https://akademosweb.uned.es/Default.aspx?id_servicio=19&modo_23=1&asigna=71022011)

Además de tutorizar los contenidos teóricos de la asignatura, los tutores serán los encargados de desarrollar las sesiones presenciales de seguimiento y control de las prácticas. Para la realización de las prácticas el alumno debe ponerse en contacto con el tutor correspondiente, para conocer cuanto antes el horario y las sesiones previstas.

- Entorno Virtual. A través de la plataforma educativa, el equipo docente de la asignatura proporcionará a los alumnos una variedad de materiales de apoyo para el estudio, así como el enunciado de la práctica obligatoria. También se ofrecen foros donde los alumnos pueden plantear sus preguntas y recibir respuestas de los tutores o del equipo docente. El entorno virtual es el soporte fundamental de la asignatura y la principal herramienta de comunicación entre el equipo docente, los tutores y los alumnos, así como entre los propios alumnos.

La modalidad y tipo de actividades que se contemplan incluye: trabajo con contenidos teórico-prácticos utilizando la bibliografía y el material complementario. Trabajo autónomo con las actividades de ejercicios y pruebas de autoevaluación disponibles, y realización de una práctica bajo la supervisión del tutor, con las herramientas y directrices preparadas por el equipo docente. La interacción con el equipo docente y los tutores se describe más adelante.

Para solicitar plaza/turno de prácticas de laboratorio/experimentales, el estudiante tendrá que acceder a la aplicación de prácticas desde su escritorio. En estas imágenes puede ver desde dónde se puede realizar el acceso a dicha aplicación:

[https://descargas.uned.es/publico/pdf/guias/ACCESO\\_PRACTICAS\\_GRADOS\\_2017.pdf](https://descargas.uned.es/publico/pdf/guias/ACCESO_PRACTICAS_GRADOS_2017.pdf)

Si al acceder a ella no encuentra ninguna oferta, deberá ponerse en contacto con el centro asociado donde está matriculado.

Finalmente, cabe destacar que el libro base de la asignatura tiene una estructura común y muy didáctica para cada patrón, lo que ayuda mucho a su comprensión. Empieza con el objetivo principal del patrón, continúa con un ejemplo de su aplicación, presenta su estructura y dominios de aplicación, y finalmente proporciona un ejemplo en Java que el alumno puede descargar y probar.

Además, existen otros recursos disponibles para los alumnos en el curso virtual de la

asignatura.

## SISTEMA DE EVALUACIÓN

### TIPO DE PRUEBA PRESENCIAL

Tipo de examen	Examen mixto
Preguntas test	10
Preguntas desarrollo	2
Duración del examen	120 (minutos)
Material permitido en el examen	

No está permitido el uso de material adicional.

### Criterios de evaluación

Para que el examen de un alumno sea calificado deberá haber asistido, como mínimo, a una sesión presencial de prácticas en su centro asociado y haber entregado y aprobado la práctica obligatoria.

**Para que un alumno pueda aprobar la asignatura deberá haber superado un mínimo de preguntas establecido en la parte teórica (tipo test) del examen.**

**El examen consiste en 10 preguntas test y dos de desarrollo. Tener la parte del test aprobado es necesario para la corrección de la parte de desarrollo. En la parte test, sólo una de las respuestas es válida. Las respuestas correctas se puntuarán con +1,0, mientras que las respondidas de manera incorrecta se puntuarán con -0,25. Las no contestadas no tendrán influencia ni positiva ni negativa en la nota.**

% del examen sobre la nota final	90
Nota del examen para aprobar sin PEC	
Nota máxima que aporta el examen a la calificación final sin PEC	
Nota mínima en el examen para sumar la PEC	
Comentarios y observaciones	

### PRUEBAS DE EVALUACIÓN CONTINUA (PEC)

¿Hay PEC?	No
Descripción	
Criterios de evaluación	
Ponderación de la PEC en la nota final	
Fecha aproximada de entrega	
Comentarios y observaciones	



**OTRAS ACTIVIDADES EVALUABLES**

¿Hay otra/s actividad/es evaluable/s? Si

Descripción

**Práctica Obligatoria**

**El trabajo del curso incluye la realización de una práctica obligatoria de programación.**

**El objetivo de la práctica es brindar al alumno la oportunidad de explorar la aplicación práctica de los patrones en un ejemplo dentro del contexto de un sistema de gestión. El tipo de sistema y los patrones que se deben utilizar varían cada año. La práctica se divide en cuatro partes, cada una con una estructura similar que incluye preguntas para que el alumno responda en la memoria de la práctica.**

Criterios de evaluación

Las prácticas son corregidas por los Tutores de los Centros Asociados, quienes las reenvían después a la sede central. La nota asignada por el tutor podrá incrementar hasta un máximo de 1 punto en la nota final de la asignatura, por supuesto, siempre que esté aprobada.

**Una vez aprobada la práctica, se guardará la nota para la convocatoria de septiembre en caso de ser necesario. El alumno podrá entregar la práctica después de la fecha establecida en mayo, siguiendo las indicaciones de su tutor.**

**En ningún momento se podrá realizar el examen sin haber aprobado la práctica.**

**Las notas de las prácticas no se guardan de un curso para otro.**

Ponderación en la nota final	1
Fecha aproximada de entrega	31/05/2018
Comentarios y observaciones	

**¿CÓMO SE OBTIENE LA NOTA FINAL?**

La ponderación final tanto de la práctica como de cada una de las partes del examen en la nota final de la asignatura quedaría de la siguiente manera:

Práctica Obligatoria: 1 punto sobre 10

Parte Teórica del Examen: 2,5 puntos sobre 10

Parte Práctica del Examen: 6,5 puntos sobre 10

**BIBLIOGRAFÍA BÁSICA**

ISBN(13):9782409016349

Título: PATRONES DE DISEÑO EN JAVA LOS 23 MODELOS DE DISEÑO: DESCRIPCIONES Y SOLUCIONES ILUSTRADAS EN UML 2 Y JAVA (2ª EDICIÓN) (2ª edición)

Autor/es: Laurent Debrauwer ;

Editorial: Ediciones ENI

Tal y como indica el editorial en su página web, el libro presenta de forma concisa y práctica los 23 modelos de diseño (Design Patterns) fundamentales ilustrándolos mediante ejemplos

adaptados y rápidos de comprender. Cada ejemplo se describe en UML y en Java bajo la forma de un pequeño programa completo y ejecutable. Para cada patrón el autor detalla su nombre, el problema correspondiente, la solución propuesta, sus dominios de aplicación y su estructura genérica.

El libro está organizado en cinco partes. En la primera, el autor introduce la noción de patrón de diseño. Las tres siguientes detallan cada una de las tres familias de patrones de diseño: los patrones de construcción, los patrones de estructuración y los patrones de comportamiento. Para terminar, la última parte presenta tres variantes de patrones existentes, mostrando la gran flexibilidad existente a la hora de implementar estos modelos. También se aborda el patrón compuesto MVC (Model-View-Controller). Esta nueva edición del libro se enriquece con un capítulo sobre los conceptos avanzados de la programación orientada a objetos, que permite al lector profundizar en sus conocimientos.

El libro está disponible en la web del editorial tanto en papel como en formato electrónico (que se puede acceder en línea) con un descuento.

## BIBLIOGRAFÍA COMPLEMENTARIA

Como material complementario, el equipo docente recomienda la consulta de los siguientes documentos:

- Patterns in Java: A Catalog of Reusable Design Patterns Illustrated with UML, 2nd Edition, Volume 1. Marc Grand. Wiley. 2002
- Pattern Hatching: Design Patterns Applied. John Vlissides. Addison-Wesley. 1998.
- Design Patterns Explained: A New Perspective on Object-Oriented Design. Alan Shalloway, James R. Trott. Addison-Wesley. 2001.

## RECURSOS DE APOYO Y WEBGRAFÍA

Como materiales adicionales para el estudio de la asignatura se ofrece en el curso virtual:

- Esta guía de estudio y una versión extendida de la misma, la guía didáctica.
- Enunciados y soluciones de ejercicios teórico-prácticos que el alumno puede usar como ejercicios de autoevaluación.
- Exámenes resueltos de anteriores convocatorias.
- Lista de preguntas frecuentes, que recogen dudas de años anteriores.

Además, a través de CiberUNED se pondrán a disposición de los alumnos herramientas necesarias para el desarrollo de la práctica. Los alumnos pueden acceder al curso virtual en este entorno con su identificador y clave de acceso. En los centros asociados los alumnos dispondrán de ordenadores en donde el entorno de desarrollo BlueJ deberá estar instalado. Además, los alumnos que dispongan de un ordenador personal podrán instalarse dicho

entorno de desarrollo. En el entorno CiberUNED se encontrará este paquete software y las instrucciones para su instalación. El entorno virtual se usará como medio para que los estudiantes puedan acceder a material complementario de estudio. El equipo docente añadirá aquellos documentos o referencias que considere adecuados para que los alumnos puedan complementar los conocimientos adquiridos a través del estudio de la bibliografía básica.

---

## **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.