

Facultad de Ciencias

## GUÍA DOCENTE DE LA ASIGNATURA

G271 - Métodos de Programación

Grado en Ingeniería Informática  
Obligatoria. Curso 1

Grado en Matemáticas  
Optativa. Curso 4

Curso Académico 2018-2019

### 1. DATOS IDENTIFICATIVOS

Título/s	Grado en Ingeniería Informática Grado en Matemáticas		Tipología y Curso	Obligatoria. Curso 1 Optativa. Curso 4
Centro	Facultad de Ciencias			
Módulo / materia	MATERIA PROGRAMACIÓN DE COMPUTADORES MENCION EN INFORMÁTICA MÓDULO OBLIGATORIO			
Código y denominación	G271 - Métodos de Programación			
Créditos ECTS	6	Cuatrimestre	Cuatrimestral (2)	
Web	<a href="http://moodle.unican.es/moodle27/course/view.php?id=460">http://moodle.unican.es/moodle27/course/view.php?id=460</a>			
Idioma de impartición	Español	English friendly	No	Forma de impartición Presencial

Departamento	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA			
Profesor responsable	MARIO ALDEA RIVAS			
E-mail	mario.aldea@unican.es			
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO - COMPUTADORES TIEMPO REAL (3054)			
Otros profesores	JOSE CARLOS PALENCIA GUTIERREZ ADOLFO GARANDAL MARTIN			

### 2. CONOCIMIENTOS PREVIOS

Haber cursado la asignatura 'Introducción al Software'

### 3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS

Competencias Genéricas
Capacidad de análisis, síntesis y evaluación.
Capacidad de organización y planificación.
Capacidad de resolución de problemas aplicando técnicas de ingeniería.
Capacidad para argumentar y justificar lógicamente las decisiones tomadas y las opiniones.
Capacidad de trabajo en equipo.
Capacidad de relación interpersonal.
Razonamiento crítico.
Aprendizaje autónomo.
Adaptación a nuevas situaciones.
Creatividad.
Tener motivación por la calidad.
Competencias Específicas
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.
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.
Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.
Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

#### 3.1 RESULTADOS DE APRENDIZAJE

- Capacidad de crear aplicaciones sencillas utilizando una metodología orientada a objetos, separando las fases de análisis, diseño e implementación
- Capacidad de codificar en Java un diseño basado en objetos
- Capacidad crear jerarquías de clases para sacar partido herencia y el polimorfismo en el desarrollo de programas
- Capacidad de crear y usar clases que utilicen excepciones
- Capacidad de utilizar módulos de programa predefinidos
- Capacidad de documentar correctamente una clase Java
- Capacidad de guardar y recuperar datos de un fichero
- Capacidad de aplicar estrategias de prueba sencillas para módulos
- Capacidad de crear y comprender diagramas UML de clases sencillos
- Conocimiento y aplicación de los algoritmos básicos de recorrido y búsqueda sobre tablas y secuencias.

#### 4. OBJETIVOS

El principal objetivo de la asignatura es que el alumno domine los principales conceptos de la metodología de programación orientada a objetos y sepa aplicarles al desarrollo de programas de mediana complejidad utilizando el lenguaje Java.

La asignatura pretende dotar al alumno de los siguientes conocimientos:

- Conocimientos básicos sobre el proceso integral de desarrollo de software
- Diferencia y relación entre los conceptos de clase, objeto y referencia.
- Conceptos de clase y tipo primitivo
- Utilización de las clases como elemento básico de estructuración del código
- Conceptos de herencia y polimorfismo
- Tratamiento de errores mediante excepciones
- Modularidad y la abstracción como principios básicos para la construcción de una aplicación.
- Utilización de tablas y secuencias: recorridos y búsquedas.
- Diferencia entre ficheros binarios y de texto.
- Técnicas básicas para la prueba de módulos

La asignatura utilizará el lenguaje Java como lenguaje de programación sobre el que concretar y practicar los conceptos estudiados.

#### 5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES

ACTIVIDADES	HORAS DE LA ASIGNATURA
<b>ACTIVIDADES PRESENCIALES</b>	
<b>HORAS DE CLASE (A)</b>	
- Teoría (TE)	20
- Prácticas en Aula (PA)	10
- Prácticas de Laboratorio (PL)	30
- Horas Clínicas (CL)	
Subtotal horas de clase	60
<b>ACTIVIDADES DE SEGUIMIENTO (B)</b>	
- Tutorías (TU)	7,5
- Evaluación (EV)	7,5
Subtotal actividades de seguimiento	15
<b>Total actividades presenciales (A+B)</b>	<b>75</b>
<b>ACTIVIDADES NO PRESENCIALES</b>	
Trabajo en grupo (TG)	20
Trabajo autónomo (TA)	55
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
<b>Total actividades no presenciales</b>	<b>75</b>
<b>HORAS TOTALES</b>	<b>150</b>

## 6. ORGANIZACIÓN DOCENTE

CONTENIDOS		TE	PA	PL	CL	TU	EV	TG	TA	TU- NP	EV- NP	Semana
1	Presentación de la asignatura	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1
2	1. Introducción al lenguaje Java. Características principales. Compilación para máquina virtual. Estructura de un programa. Datos y expresiones. Estructuras algorítmicas. Datos compuestos. Entrada/salida básica. Concepto de clase y objeto. Introducción a las clases y objetos en Java.	2,00	2,00	4,00	0,00	0,75	1,00	3,00	7,00	0,00	0,00	1,2,3
3	2. Proceso de desarrollo de software. Ingeniería del software. Procesos de desarrollo de software. Modelado del software: UML. Introducción al análisis de requisitos. Diseño orientado a objetos. Programación orientada a objetos. Proceso de desarrollo usado en la asignatura.	2,00	0,00	2,00	0,00	0,75	0,50	1,00	4,00	0,00	0,00	2,3
4	3. Clases, referencias y objetos en Java. Creación e inicialización de objetos. Tipos primitivos, referencias y objetos. Comparación de objetos. Recolector de basura. Métodos y campos de clase (o estáticos). Anidamiento de clases.	3,00	1,00	4,00	0,00	1,00	1,00	2,50	7,00	0,00	0,00	5,6
5	4. Diseño Modular. Conceptos de modularidad y abstracción. Modificadores de acceso básicos. Paquetes. Módulos genéricos. Programación con módulos predefinidos. Documentación del código fuente. Patrón Modelo-Vista-Controlador.	2,00	1,00	4,00	0,00	0,75	1,00	2,50	6,00	0,00	0,00	7,8
6	5. Herencia y Polimorfismo. Herencia. Clases abstractas. Polimorfismo. La clase Object.	3,00	2,00	4,00	0,00	1,50	1,00	3,00	7,50	0,00	0,00	8,9
7	6. Tratamiento de errores. Tratamiento de errores por paso de parámetros. Excepciones Java: generación y propagación automática. Bloques de tratamiento excepciones. La cláusula finally. Patrones de tratamiento de excepciones. Lanzar excepciones. Creación de excepciones propias. Excepciones "comprobadas" y cláusula throws. Notificación de errores mediante excepciones. Usos incorrectos de las excepciones.	2,00	1,00	4,00	0,00	0,75	1,00	2,50	7,00	0,00	0,00	10,11
8	7. Entrada/salida con ficheros. Conceptos básicos. Flujos de datos (streams). E/S de texto. E/S de texto con formato. E/S binaria.	3,00	1,00	4,00	0,00	1,00	1,00	2,50	8,00	0,00	0,00	12, 13
9	8. Prueba de programas. Verificación y validación. Pruebas del software. Caja negra. Herramienta JUnit.	1,00	1,00	2,00	0,00	0,50	0,50	1,50	3,50	0,00	0,00	14
10	Revisión general de la asignatura	1,00	1,00	2,00	0,00	0,50	0,50	1,50	5,00	0,00	0,00	15

TOTAL DE HORAS	20,00	10,00	30,00	0,00	7,50	7,50	20,00	55,00	0,00	0,00	
Esta organización tiene carácter orientativo.											

TE	Horas de teoría
PA	Horas de prácticas en aula
PL	Horas de prácticas de laboratorio
CL	Horas Clínicas
TU	Horas de tutoría
EV	Horas de evaluación
TG	Horas de trabajo en grupo
TA	Horas de trabajo autónomo
TU-NP	Tutorías No Presenciales
EV-NP	Evaluación No Presencial

## 7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Prácticas	Evaluación en laboratorio	No	No	5,00
Calif. mínima	0,00			
Duración	2			
Fecha realización	A lo largo del curso			
Condiciones recuperación				
Observaciones	<p>La nota se obtiene en base a:</p> <ul style="list-style-type: none"> <li>- Presentación de las prácticas: a lo largo del curso el profesor requerirá al alumno la presentación de alguna de las prácticas realizadas.</li> <li>- Informes de prácticas: deberán entregarse dentro del plazo establecido. Una entrega fuera del plazo supondrá una penalización en la nota. También supondrá una penalización la entrega de una práctica con errores u omisiones graves.</li> </ul>			
Exámenes prácticos	Evaluación en laboratorio	No	Sí	55,00
Calif. mínima	4,50			
Duración	2 horas			
Fecha realización	A lo largo del curso			
Condiciones recuperación	Examen de prácticas en el periodo ordinario y examen final en el periodo de recuperación			
Observaciones	<p>Exámenes realizados en el laboratorio. Está permitido llevar apuntes.            Se celebrarán entre 2 y 4 exámenes de prácticas a lo largo del curso.            La nota de este apartado se calculará como la media (ponderada) de las notas de los exámenes realizados. Lo normal será que el peso de todos los exámenes sea el mismo, aunque podría darse el caso de primar unos sobre otros si se considera que hay razones para ello.            La nota de los exámenes prácticos podrá verse mejorada si el alumno realiza satisfactoriamente los problemas y ejercicios que podrían proponerse a lo largo del curso.</p>			
Examen final	Examen escrito	Sí	Sí	40,00
Calif. mínima	4,50			
Duración	3,5 horas			
Fecha realización	Periodo ordinario de exámenes			
Condiciones recuperación	Examen final en el periodo de recuperación			
Observaciones	<p>Compuesto por varias cuestiones y uno o más problemas.            Las cuestiones son preguntas cortas que pretenden evaluar los conocimientos del alumno en alguno de los contenidos de la asignatura.            Los problemas pretenden evaluar el dominio de la metodología de programación que tiene el alumno. Para ello se planteará el desarrollo de una aplicación similar a las realizadas en prácticas.            Se podrán utilizar apuntes y bibliografía.            Para poder presentarse a este examen es obligatorio entregar y presentar las prácticas de la asignatura.</p>			
<b>TOTAL</b>				<b>100,00</b>
<b>Observaciones</b>				
<p>Para aprobar la asignatura es necesario superar la nota mínima de los 'Exámenes prácticos' y del 'Examen final'. En caso de no superar alguna de esas notas mínimas, la nota final será el mínimo de 4.5 y la media obtenida.</p> <p>Para poder presentarse a los exámenes finales (escrito y de prácticas) de los periodos ordinario y de recuperación es obligatorio entregar y presentar las prácticas de la asignatura.</p>				
<b>Criterios de evaluación para estudiantes a tiempo parcial</b>				

En el caso de estudiantes en regímenes de dedicación a tiempo parcial que no puedan realizar alguno de los exámenes prácticos podrán recuperar dichos exámenes en el examen final de prácticas del periodo ordinario y/o en el examen final en el periodo de recuperación.

Para poder realizar los exámenes finales (escrito y de prácticas) de los periodos ordinarios y de recuperación será obligatorio que, con anterioridad, hayan entregado y presentado las prácticas.

Se espera de estos alumnos que vayan realizando las prácticas de la asignatura al ritmo que se van proponiendo y que las vayan entregando en los plazos que a tal efecto se establezcan. La entrega fuera de plazo de las prácticas les supondrá la misma penalización que a los alumnos que asisten habitualmente a las clases de la asignatura.

### 8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

#### BÁSICA

Ken Arnold, James Gosling, David Holmes, "El lenguaje de programación Java", 3ª edición. Addison-Wesley, 2000.

Francisco Gutiérrez, Francisco Durán, Ernesto Pimentel. "Programación Orientada a Objetos con Java". Paraninfo, 2007.

#### Complementaria

D.J. Barnes y M. Kölling, "Objects first with Java: A practical introduction using Bluej", Prentice Hall, 2003.

Ian Sommerville, "Ingeniería de software" (9a edición). Pearson Educación de México, 2011.

King, Kim N. "Java programming: from the beginning". W. W. Norton & Company, cop. 2000

The Java Tutorials. <http://docs.oracle.com/javase/tutorial/>

Eitel, Harvey M. y Deitel, Paul J., "Cómo programar en Java", 9ª edición. Pearson Educación, Mexico, 2012.

### 9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Entorno de desarrollo Eclipse	Facultad de Ciencias			
Sistema operativo Linux	Facultad de Ciencias			
Compilador Java de Sun, versión 8	Facultad de Ciencias			

### 10. COMPETENCIAS LINGÜÍSTICAS

- |   |   |
|---|---|
| <input type="checkbox"/> Comprensión escrita                            | <input type="checkbox"/> Comprensión oral |
| <input type="checkbox"/> Expresión escrita                              | <input type="checkbox"/> Expresión oral   |
| <input type="checkbox"/> Asignatura íntegramente desarrollada en inglés |   |

#### Observaciones