

Facultad de Ciencias

Grado en Ingeniería Informática

## GUÍA DOCENTE DE LA ASIGNATURA

G266 - Introducción al Software

Curso Académico 2013-2014

### 1. DATOS IDENTIFICATIVOS DE LA ASIGNATURA

Título/s	Grado en Ingeniería Informática
Centro	Facultad de Ciencias
Módulo / materia	ASIGNATURAS DE PRIMER CURSO MATERIA FUNDAMENTOS DE INFORMÁTICA
Código y denominación	G266 - Introducción al Software
Créditos ECTS	6
Curso / Cuatrimestre	CUATRIMESTRAL (1)
Web	<a href="http://www.ctr.unican.es/asignaturas/intro_sw">http://www.ctr.unican.es/asignaturas/intro_sw</a>
Idioma de impartición	Español
Forma de impartición	Presencial

Departamento	DPTO. MATEMATICAS, ESTADISTICA Y COMPUTACION
Profesor responsable	MICHAEL GONZALEZ HARBOUR
E-mail	michael.gonzalez@unican.es
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO PROFESORES (3055)
Otros profesores	ADOLFO GARANDAL MARTIN DIEGO GARCIA SAIZ CARLOS BLANCO BUENO PABLO SANCHEZ BARREIRO

### 2. CONOCIMIENTOS PREVIOS

Los propios del bachillerato.

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

Competencias Genéricas	Nivel
Capacidad de organización y planificación.	1
Capacidad de resolución de problemas aplicando técnicas de ingeniería.	1
Capacidad para argumentar y justificar lógicamente las decisiones tomadas y las opiniones.	1
Capacidad de relación interpersonal.	1
Razonamiento crítico.	1
Aprendizaje autónomo.	1
Creatividad.	1
Tener motivación por la calidad.	1
Competencias Específicas	Nivel
Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.	3

### 3.1 RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA

- Conocer y comprender la sintaxis y semántica de las expresiones e instrucciones básicas de un lenguaje de programación imperativo
- Ser capaz de diseñar, implementar y probar algoritmos y programas sencillos en un lenguaje imperativo estructurado de alto nivel, y entender los razonamientos sobre su comportamiento.
- Programar procedimientos y funciones: aplicar el estilo modular en diseño de código de alto nivel y conocer las técnicas de documentación y desarrollo de aplicaciones.
- Utilizar un entorno de programación para editar, compilar y ejecutar programas
- Entender los principios comunes de claridad y precisión requeridos por las tareas de programación en todos los niveles.
- Saber usar sistemas operativos para realizar tareas básicas
- Saber usar una base de datos para almacenar y recuperar información tabular
- Saber usar una hoja de cálculo para realizar cálculos sencillos sobre tablas de datos

### 4. OBJETIVOS DE LA ASIGNATURA

Alcanzar los resultados del aprendizaje

### 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)	10
- Evaluación (EV)	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 DE LA ASIGNATURA												
CONTENIDOS		TE	PA	PL	CL	TU	EV	TG	TA	TU- NP	EV- NP	Semana
1	BLOQUE TEMÁTICO 1: Programación en un lenguaje orientado a objetos	0,00	0,00	0,00	0,00	6,00	3,00	15,00	40,00	0,00	0,00	Todas
1.1	0. 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
1.2	1. Introducción a los lenguajes de programación. Lenguajes de alto nivel. El proceso de compilación. El ciclo de vida del software. Concepto de algoritmo. Concepto de clase y objeto. Diagramas de clases. Estructura de un programa. Estructura de un método.	2,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1-2
1.3	2. Datos y expresiones. Tipos primitivos. Variables y constantes. Operadores y expresiones. Conversión de tipos. Uso de funciones matemáticas. Declaración de objetos. Strings. Composición de objetos. Atributos y métodos estáticos.	4,00	3,00	6,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	2-4
1.4	3. Estructuras algorítmicas. Instrucción condicional. Instrucción condicional múltiple. Instrucciones de bucle. Recursión. Descripción de algoritmos mediante pseudocódigo.	4,00	2,00	6,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	3-6
1.5	4. Estructuras de datos elementales. Arrays y tablas unidimensionales. Algoritmos de recorrido y búsqueda. Arrays multidimensionales. Tipos enumerados	3,00	2,00	4,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	6-8
1.6	5. Entrada/salida. Entrada/salida de texto y de caracteres. Entrada/salida de números. Gráficas. Dibujos. Menús de botones.	2,00	1,00	4,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	3-10
2	BLOQUE TEMÁTICO 2: Herramientas	0,00	0,00	0,00	0,00	4,00	2,00	5,00	15,00	0,00	0,00	Todas
2.1	6. Uso de sistemas operativos. Conceptos básicos. Sistemas operativos comunes. Arranque y finalización de sesiones. El sistema de ficheros. El intérprete de órdenes. Ejecución de programas. El gestor gráfico de ficheros. Uso de la memoria USB. Guiones (scripts). Aspectos avanzados.	1,00	0,00	2,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1-3
2.2	7. Uso de un entorno integrado de desarrollo de programas. Proceso de desarrollo de programas. El compilador y la ejecución. Entorno integrado de desarrollo. La depuración. Generación de documentos. Empaquetamiento del programa.	0,00	0,00	2,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	2-3
2.3	8. Uso de una hoja de cálculo. Introducción. Creación de una hoja de cálculo. Fórmulas. Formatear las celdas. Gráficos. Hojas múltiples. Ordenar datos	1,00	1,00	2,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	9-10
2.4	9. Uso de una base de datos. Introducción. Creación de una base de datos. Tablas de datos. Relaciones entre datos. Consultas. Formularios. Informes.	2,00	1,00	4,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	10-11

TOTAL DE HORAS	20,00	10,00	30,00	0,00	10,00	5,00	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.	%
Teoría: participación en clase	Otros	No	No	10,00
Calif. mínima	0,00			
Duración	Continua			
Fecha realización	Continua			
Condiciones recuperación				
Observaciones	Teoría: Se realizará una evaluación continuada basada en la participación en clase			
Problemas	Actividad de evaluación con soporte virtual	No	Sí	10,00
Calif. mínima	0,00			
Duración	Continua			
Fecha realización	Continua			
Condiciones recuperación	Examen			
Observaciones	Problemas: Se realizará una evaluación continuada de problemas propuestos y resueltos en casa. En cada problema se requerirá una presentación al grupo, así como una breve memoria. En cada sesión habrá ejercicios evaluables y otros no evaluables.			
Examen final	Examen escrito	Sí	Sí	50,00
Calif. mínima	4,00			
Duración	3.5 horas			
Fecha realización	Febrero y Septiembre (Periodo ordinario y periodo de recuperación)			
Condiciones recuperación	Examen			
Observaciones	Examen final: Se realizarán dos exámenes escritos de cuestiones y problemas: uno en el periodo de evaluación ordinario y uno en el periodo de recuperación. Duración: 3.5 horas, en dos partes: una de cuestiones teórico-prácticas, y la otra para resolver un problema de programación. Se pueden llevar apuntes y libros a ambas partes. Cada parte es la mitad de la nota del examen.			
Prácticas	Evaluación en laboratorio	No	Sí	30,00
Calif. mínima	4,00			
Duración	Continua			
Fecha realización	Continua			
Condiciones recuperación	Examen			

Observaciones	<p><b>Prácticas:</b> Se realizarán tres evaluaciones de prácticas durante el transcurso de la asignatura; estas evaluaciones consistirán en la realización de una práctica en el laboratorio y la entrega de un breve informe al finalizar la sesión; la media de estas tres calificaciones será la nota de prácticas; en el resto de las prácticas se requerirá una breve memoria. Para poder aprobar las prácticas en el periodo ordinario se requiere haber entregado al menos todas las memorias de las prácticas excepto una, antes de la tercera práctica evaluable.</p> <p>Los criterios en las evaluaciones de prácticas serán:</p> <p>a) Trabajo en el laboratorio</p> <ul style="list-style-type: none"> <li>• Conocimientos: Facilidad para diseñar y nivel de comprensión de los conceptos</li> <li>• Herramientas: Habilidad en el manejo de las herramientas utilizadas en el laboratorio.</li> <li>• Grado de Resolución: Grado de resolución de la práctica alcanzado durante las horas en el laboratorio.</li> </ul> <p>b) Informes sobre las prácticas</p> <ul style="list-style-type: none"> <li>• Estilo de programación: Estilo, claridad, documentación.</li> <li>• Organización: Organización del informe, claridad en la exposición</li> <li>• Plazo de entrega. Por los retrasos en la entrega de los informes de las prácticas previas a cada práctica evaluable habrá una penalización: 0.5 puntos por cada entrega retrasada hasta 1 semana, y 1 punto por cada entrega retrasada hasta 2 semanas.</li> </ul>
TOTAL	100,00
Observaciones	
La evaluación continua de problemas y prácticas se podrá superar en el periodo de recuperación mediante un examen de prácticas: 40%	
Será un examen en el laboratorio, de 3 horas de duración, en el que habrá que resolver un ejercicio práctico. Se pueden llevar apuntes y libros. Para poder presentarse a este examen se deberán entregar las memorias de todas las prácticas obligatorias de la asignatura.	
Para superar la asignatura es preciso superar tanto el examen final como los problemas y prácticas.	
En caso de que una de estas partes no se supere, la nota final será el mínimo de 4.5 y la media obtenida.	
En caso de aprobar únicamente una de las dos partes en el periodo ordinario, se guardaría la nota de esa parte para el periodo de recuperación.	
Observaciones para alumnos a tiempo parcial	
La evaluación continua de problemas y prácticas se podrá superar por los estudiantes a tiempo parcial mediante un examen de prácticas (40%), tanto en el periodo ordinario como en el periodo de recuperación: Para estos alumnos el peso del examen de teoría será del 60%.	
Será un examen en el laboratorio, de 3 horas de duración, en el que habrá que resolver un ejercicio práctico. Se pueden llevar apuntes y libros. Para poder presentarse a este examen se deberán entregar las memorias de todas las prácticas obligatorias de la asignatura.	

## 8. BIBLIOGRAFIA

### BÁSICA

- D.J. Barnes y M. Kölling, "Objects first with Java: A practical introduction using Bluej", Prentice Hall, 2003.
- David M. Arnow, Gerald Weiss. "Introducción a la programación con Java: Un enfoque orientado a objetos". Prentice Hall, 2000. ISBN 84-7829-033-8
- Ken Arnold, James Gosling, David Holmes, "El lenguaje de programación Java", 3ª edición. Addison-Wesley, 2001, ISBN 84-7829-045-1
- Bertrand Meyer. "Construcción Software Orientada a Objetos". Prentice Hall. 2ª Edición, Prentice Hall, 2000.
- Cursillos de introducción a Open Office:  
<http://documentation.openoffice.org/manuals/oooauthors2/index.html>

### Complementaria

- J. Castro, F. Cucker, X. Messeger, A. Rubio L. Solano, B. Valles. "Curso de Programacion". McGraw- Hill, 1993
- S. Zakhour, S. Hommel, J. Royal, I. Rabinovitch, T. Risser, M. Hoeber, "The Java Tutorial Fourth Edition: A Short Course on the Basics". Pearson Education, 2006
- The Java Tutorials. <http://java.sun.com/docs/books/tutorial/>
- Liskov, B., Guttag, J. "Abstraction and Specification in Program Development", McGraw-Hill, 1986.
- Patrick Naughton, Herbert Schildt, Herb Schildt. "Java 2, The complete reference". Osborne McGraw-Hill 1999. ISBN: 0072119764
- E. Bueno, y otros. "Algoritmos y Ejercicios resueltos en Java". Prentice Hall 2003. ISBN 84-705-4024-2
- Edsger W. Dijkstra. "A Discipline of Programming". Prentice Hall, 1976.
- Michael A. Jackson. "Principles of Program Design". Academic Press, 1975.
- Jesús Sánchez Allende y otros. "Programación en Java". 3ª edición. Mc Graw Hill, 2009

## 9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Entorno de desarrollo Bluej versión 3.0.9 o superior, con compilador Java de Oracle versión 7 o superior, y sistema operativo Linux que incluya shell bash, descompresor zip, editor de texto gedit. Editor de texto Emacs. Open Office Write, Calc y Base				



#### 10. COMPETENCIAS LINGÜÍSTICAS

- |   |   |
|---|---|
| <input checked="" 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**

Lectura de documentación técnica en inglés