



Programación II

Guía de Aprendizaje – Información al estudiante

1. Datos Descriptivos

Asignatura	Programación II
Materia	Programación
Departamento responsable	DLSIIS
Créditos ECTS	6
Carácter	Obligatoria
Titulación	Graduado en Informática
Curso	2011-2012
Especialidad	No aplica

Curso académico	2011-2012
Semestre en que se imparte	Ambos (septiembre a enero y febrero a junio)
Semestre principal	Segundo
Idioma en que se imparte	Castellano
Página Web	Moodle



2. Profesorado

NOMBRE Y APELLIDO	DESPACHO	Correo electrónico
Raúl García Castro (Coord.)	2302	rgarcia@fi.upm.es
Ángel Lucas González	2310/CETTICO	agonzalez@fi.upm.es
Jaime Ramírez	5112	jramirez@fi.upm.es
Raúl Alonso	2307	ralonso@fi.upm.es
Clara Benac	2308	cbenac@fi.upm.es
Susana Muñoz	2310	susana@fi.upm.es

3. Conocimientos previos requeridos para poder seguir con normalidad la asignatura

Asignaturas superadas	Programación I
Otros resultados de aprendizaje necesarios	Conocimientos de la sintaxis básica de Java: sentencias básicas, tipos básicos, sentencias de control de flujo y definición de funciones.

4. Objetivos de Aprendizaje

COMPETENCIAS ASIGNADAS A LA ASIGNATURA Y SU NIVEL DE ADQUISICIÓN		
Código	Competencia	Nivel
CE-3/4	Capacidad de elegir y usar los métodos analíticos y de modelización relevantes, y de describir una solución de forma abstracta.	3
CE-6	Comprender intelectualmente el papel central que tienen los algoritmos y las estructuras de datos, así como una apreciación del mismo.	1
CE-8	Poseer destrezas fundamentales de la programación que permitan la implementación de los algoritmos y las estructuras de datos en el software.	2
CE-9	Poseer las destrezas que se requieren para diseñar e implementar unidades estructurales mayores que utilizan los algoritmos y las estructuras de datos, así como las interfaces por las que se comunican estas unidades.	3
CG-1/21	Capacidad de resolución de problemas aplicando conocimientos de matemáticas, ciencias e ingeniería.	G1
CG-2/CE45	Capacidad para el aprendizaje autónomo y la actualización de conocimientos, y reconocimiento de su necesidad en el área de la informática.	G1
CG-3/4	Saber trabajar en situaciones de falta de información y bajo presión, teniendo nuevas ideas, siendo creativo.	G1
CG-6	Capacidad de abstracción, análisis y síntesis.	G1
CG-7/8/9/10/16/17	Capacidad para trabajar dentro de un equipo, organizando, planificando, tomando decisiones, negociando y resolviendo conflictos, relacionándose, y criticando y haciendo autocrítica.	G1

LEYENDA: Nivel de adquisición 1: Conocimiento
 Nivel de adquisición 2: Comprensión
 Nivel de adquisición 3: Aplicación
 Nivel de adquisición 4: Análisis y síntesis
 Nivel de adquisición G1: Básico
 Nivel de adquisición G2: Medio
 Nivel de adquisición G3: Alto



RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA			
Código	Resultado de aprendizaje	Competencias asociadas	Nivel de adquisición
RA1	Traducir especificaciones de tipos abstractos de datos (TADs) a implementaciones Java (p.ej.) correctas.	CE-4, CE-6, CE-8, CE-9	3
RA2	Programar aplicaciones mediante librerías existentes de TADs, iteradores, etc., extendiendo su funcionalidad (con herencia) o adaptándolas a un uso particular (instanciación de genéricos).	CE-8, CE-9	3
RA3	Documentar clases y bibliotecas, tanto de manera pública (hacia el cliente) como privada (hacia el implementador).	CE-4, CE-9	3
RA4	Realizar pruebas para asegurar el correcto funcionamiento de un TAD así como su integración en la aplicación que lo usa.	CE-8, CE-9	3

5. Sistema de evaluación de la asignatura

INDICADORES DE LOGRO		
Ref	Indicador	Relacionado con RA
I1	Implementar un TAD a partir del diseño del mismo	RA1
I2	Especificar en lenguaje natural los contratos asociados a los servicios de un TAD	RA1
I3	Enumerar las ventajas de la abstracción y la modelización en el diseño de un TAD	RA1
I4	Implementar un TAD lineal (pila, cola, etc.) utilizando programación orientada a objetos	RA1
I5	Implementar una estructura de datos dinámica lineal	RA1
I6	Enumerar las ventajas de separar la interfaz de la implementación en un diseño modular	RA1
I7	Localizar y seleccionar las librerías más apropiadas para la aplicación que estemos desarrollando	RA2
I8	Implementar una clase extendiendo otra clase ya existente por medio del mecanismo de la herencia	RA2
I9	Utilizar una clase genérica en una aplicación	RA2
I10	Implementar una clase genérica que implemente un TAD dado	RA2
I11	Documentar adecuadamente la interfaz de un TAD, así como su implementación	RA2
I12	Utilizar una herramienta para la generación de la documentación del código fuente	RA3
I13	Utilizar un <i>framework</i> para la automatización de pruebas	RA3
I14	Definir el concepto de prueba de software	RA4
I15	Enumerar los distintos tipos de pruebas de software	RA4

INDICADORES DE LOGRO		
Ref	Indicador	Relacionado con RA
I16	Diseñar casos de prueba de forma que se asegure una cobertura razonable de los servicios de un TAD	RA4
I17	Implementar programas que manejen excepciones	RA1
I18	Recoge la información significativa que necesita par resolver los problemas en base a datos y no solo a opiniones subjetivas y sigue un método lógico de análisis de la información	RA1-RA4
I19	Sigue un método lógico para identificar las partes implicadas en el problema	RA1-RA4
I20	Diseña un plan de acción para la aplicación de la solución escogida	RA1-RA4
I21	Pone en practica de forma disciplinada los enfoques, métodos y experiencias que propone el profesor	RA1-RA4
I22	Pregunta para aprender y se interesa para aclarar las dudas	RA1-RA4
I23	Usa la información dada como un medio para generar ideas	RA1-RA4
I24	Percibe la información o la situación desde perspectivas diferentes	RA1-RA4
I25	Agrupar y describe conjuntos de elementos cualitativos en categorías preestablecidas	RA1-RA4
I26	Describe correctamente procesos secuenciales	RA1-RA4
I27	Toma decisiones acertadas, basadas en información objetiva	RA1-RA4
I28	Demuestra método y sistemática a la hora de tomar decisiones	RA1-RA4
I29	Colabora en la definición, organización y distribución de las tareas de grupo. Se orienta a la consecución de acuerdos y objetivos comunes y se compromete a ellos	RA1-RA4



EVALUACION CONTINUA SUMATIVA			
Breve descripción de las actividades evaluables	Momento¹	Lugar	Peso en la calif.
Examen Parcial 1 (test)	Semana 10	Aula	15%
Examen Parcial 2 (test)	Semana 16	Aula	15%
Ejercicios prácticos	Semanas 1-16	Sala Informática	20%
Prácticas	Semanas 1-16	Fuera del aula	50%
			Total: 100%

EVALUACION CON EXAMEN FINAL SUMATIVA			
Breve descripción de las actividades evaluables	Momento¹	Lugar	Peso en la calif.
Examen Final (teoría y ejercicios prácticos)	Fecha estipulada por el centro	Aula	50%
Prácticas	Semanas 1-16	Fuera del aula	50%
			Total: 100%

¹ Las fechas son aproximadas y por lo tanto orientativas

CRITERIOS DE CALIFICACIÓN

Con el fin de superar esta asignatura, el alumno debe seguir uno de estos dos itinerarios alternativos: **basado evaluación continua o basado en examen final. El alumno elegirá al principio del curso el itinerario deseado y una vez hecha la elección no podrá cambiar de itinerario.**

1. **Itinerario basado en evaluación continua:** se aplicará un esquema de **evaluación continua combinando exámenes parciales de tipo test, ejercicios prácticos que se realizarán en el aula y prácticas que se realizarán fuera del aula**, realizándose todos ellos a lo largo del semestre. Siguiendo este esquema, la nota final (NF) de la asignatura se obtendrá a partir de una nota de teoría (NT), una nota de ejercicios prácticos (NE) y una nota de prácticas (NP) mediante la siguiente fórmula:

$$NF = 0.3NT + 0.5NP + 0.2NE, \text{ si } NP \geq 5 \text{ y } NT \geq 4 \text{ y } NE \geq 3$$
$$NF = 0, \text{ e.o.c.}$$

En donde:

NT. Nota de teoría: La nota NT será un valor numérico entre 0 y 10.

- Esta nota se obtiene mediante la realización de varios tests o parciales en el periodo lectivo.
- Para poder aprobar la NT durante el periodo lectivo, se deberá obtener al menos un 3 en cada test.
- **La nota de teoría sólo se guarda de un periodo a otro del mismo curso** si esta nota es ≥ 5 . En ningún caso las notas se guardarán de un curso a otro.

NP. Nota de prácticas: La nota NP será un valor numérico entre 0 y 10.

- Esta nota se obtiene de la media de las calificaciones de varias prácticas a realizar fuera del horario de clase.
- Será obligatorio **obtener la nota mínima especificada en el enunciado para cada práctica** propuesta para poder aprobar la NP.
- Los enunciados de las prácticas, sus respectivas fechas de entrega y su peso sobre la nota NP se publicarán durante el curso.
- El código fuente entregado debe compilar en la **versión 1.6 de java**.

NE. Nota de ejercicios prácticos: La nota NE será un valor numérico entre 0 y 10.

- Esta nota se obtiene de la media de las calificaciones de varios ejercicios prácticos.
- Los ejercicios prácticos solo se realizarán en las horas de clase estipuladas para ello. **Si un alumno no entrega al menos un 80% de estos ejercicios, no podrá aprobar la asignatura en este itinerario.**
- **La nota de ejercicios prácticos sólo se guarda de un periodo a otro del mismo curso** si esta nota es ≥ 5 . En ningún caso las notas se guardarán de un curso a otro. Si la nota es menor que cinco, en el examen extraordinario deberá realizar el problema del examen y sacar en él más de un cinco.

2. **Itinerario basado en un examen final:** en este itinerario el alumno no estará obligado a asistir a clase, y por lo tanto **no tendrá que entregar los ejercicios prácticos que se propongan para ser realizados en horario de clase**. Además, **su nota de teoría la obtendrá mediante la realización de un solo examen final que cubrirá toda la materia y que constará de diversos ejercicios teóricos (NT) y ejercicios prácticos (NE)**. Por otro lado, el alumno sí que estará obligado a aprobar las prácticas que se propongan para realizar fuera del horario de clase. Siguiendo este esquema, la nota final (NF) de la asignatura se obtendrá a partir de una nota de teoría (NT) y una nota de prácticas (NP) y una nota de ejercicios prácticos (NE) mediante la siguiente fórmula:

$$NF = 0.3NT + 0.5NP + 0.2NE, \text{ si } NP \geq 5 \text{ y } NT \geq 4 \text{ y } NE \geq 3$$
$$NF = 0, \text{ e.o.c.}$$

El Sistema de evaluación mediante sólo prueba final sólo se ofrecerá si así lo exige la Normativa Reguladora de los Sistemas de Evaluación en la UPM que esté vigente en el curso académico 2011-2012, y el procedimiento para optar por este sistema estará sujeto a lo que establezca en su caso Jefatura de Estudios de conformidad con lo que estipule dicha Normativa.

En el caso de que el alumno no apruebe la asignatura siguiendo uno de estos dos itinerarios, dispondrá de la convocatoria extraordinaria.

En esta convocatoria, **su nota final se obtendrá utilizando la misma fórmula que en el itinerario basado en un examen final**. Si el alumno ya hubiera aprobado (nota igual o superior a 5) las prácticas o la teoría (tests de teoría) y ejercicios prácticos (problemas planteados en el examen) en uno de los dos itinerarios anteriores, se le conservará dicha nota, y solo tendrá que aprobar en esta convocatoria las partes de la asignatura (teoría, ejercicios prácticos o prácticas) que tuviera suspensas.



6. Contenidos y Actividades de Aprendizaje

CONTENIDOS ESPECÍFICOS		
Bloque / Tema / Capítulo	Apartado	Indicadores Relacionados
Tema 1: Introducción a la POO con el lenguaje java	1.1 Definición de clases y objetos	I4, I5, I18 – I29
	1.2 Programación modular: paquetes y visibilidad	I3, I4, I6, I7, I18 – I29
	1.3 Manejo de excepciones	I16, I18 – I29
	1.4 POO avanzada: herencia y genéricos	I7- I10, I18 – I29
	1.5 Pruebas de programas	I13 - I15, I18 – I29
	1.6 E/S en Java	I4, I7, I18 – I29
Tema 2: Tipos abstractos de datos lineales	2.1 Concepto de TAD y aplicación de los TADs para la resolución de problemas.	I2, I3, I7, I11, I12, I18 – I29
	2.2 Diseño e implementación de un TAD	I1 - I5, I18 – I29

7. Breve descripción de las modalidades organizativas utilizadas y de los métodos de enseñanza empleados

Tabla 7. Modalidades organizativas de la enseñanza

MODALIDADES ORGANIZATIVAS DE LA ENSEÑANZA		
Escenario	Modalidad	Finalidad
	Clases Teóricas	<i>Hablar a los estudiantes</i>
	Seminarios-Talleres	<i>Construir conocimiento a través de la interacción y la actividad de los estudiantes</i>
	Clases Prácticas	<i>Mostrar a los estudiantes cómo deben actuar</i>
	Prácticas Externas	<i>Completar la formación de los alumnos en un contexto profesional</i>
	Tutorías	<i>Atención personalizada a los estudiantes</i>
	Trabajo en grupo	<i>Hacer que los estudiantes aprendan entre ellos</i>
	Trabajo autónomo	<i>Desarrollar la capacidad de autoaprendizaje</i>

Tabla 5. Métodos de enseñanza

MÉTODOS DE ENSEÑANZA		
	Método	Finalidad
	Método Expositivo/Lección Magistral	Transmitir conocimientos y activar procesos cognitivos en el estudiante
	Estudio de Casos	Adquisición de aprendizajes mediante el análisis de casos reales o simulados
	Resolución de Ejercicios y Problemas	Ejercitar, ensayar y poner en práctica los conocimientos previos
	Aprendizaje Basado en Problemas (ABP)	Desarrollar aprendizajes activos a través de la resolución de problemas
	Aprendizaje orientado a Proyectos	Realización de un proyecto para la resolución de un problema, aplicando habilidades y conocimientos adquiridos
	Aprendizaje Cooperativo	Desarrollar aprendizajes activos y significativos de forma cooperativa
	Contrato de Aprendizaje	Desarrollar el aprendizaje autónomo

Se conoce como método expositivo "la presentación de un tema lógicamente estructurado con la finalidad de facilitar información organizada siguiendo criterios adecuados a la finalidad pretendida". Esta metodología -también conocida como lección (lecture)- se centra fundamentalmente en la exposición verbal por parte del profesor de los contenidos sobre la materia objeto de estudio. El término "lección magistral" se suele utilizar para denominar un tipo específico de lección impartida por un profesor en ocasiones especiales.

Análisis intensivo y completo de un hecho, problema o suceso real con la finalidad de conocerlo, interpretarlo, resolverlo, generar hipótesis, contrastar datos, reflexionar, completar conocimientos, diagnosticarlo y, en ocasiones, entrenarse en los posibles procedimientos alternativos de solución.

Situaciones en las que se solicita a los estudiantes que desarrollen las soluciones adecuadas o correctas mediante la ejercitación de rutinas, la aplicación de fórmulas o algoritmos, la aplicación de procedimientos de transformación de la información disponible y la interpretación de los resultados. Se suele utilizar como complemento de la lección magistral.

Método de enseñanza-aprendizaje cuyo punto de partida es un problema que, diseñado por el profesor, el estudiante ha de resolver para desarrollar determinadas competencias previamente definidas.

Método de enseñanza-aprendizaje en el que los estudiantes llevan a cabo la realización de un proyecto en un tiempo determinado para resolver un problema o abordar una tarea mediante la planificación, diseño y realización de una serie de actividades, y todo ello a partir del desarrollo y aplicación de aprendizajes adquiridos y del uso efectivo de recursos.

Enfoque interactivo de organización del trabajo en el aula en el cual los alumnos son responsables de su aprendizaje y del de sus compañeros en una estrategia de corresponsabilidad para alcanzar metas e incentivos grupales. Es tanto un método, a utilizar entre otros, como un enfoque global de la enseñanza, una filosofía.

Un acuerdo establecido entre el profesor y el estudiante para la consecución de unos aprendizajes a través de una propuesta de trabajo autónomo, con una supervisión por parte del profesor y durante un periodo determinado. En el contrato de aprendizaje es básico un acuerdo formalizado, una relación de contraprestación recíproca, una implicación personal y un marco temporal de ejecución.

8. Recursos didácticos

RECURSOS DIDÁCTICOS	
	Material didáctico proporcionado por los profesores
BIBLIOGRAFÍA	J. Chase, J. Lewis (2006) Estructura de Datos con Java. Pearson Educación
	L. Craig (2004) UML y Patrones. Prentice Hall
	N. Dale, H.M. Walker (1996) Abstract data types: specifications, implementations, and applications. D. C. Heath and Company
	B. Eckel (2006) Thinking In Java 4th edition. Prentice Hall
	J. L. Fuertes, A. L. González (2007) Fundamentos de la programación en Java. Koobeht
	M.T. Goodrich, R. Tamassia (2005) Data Structures and Algorithms in Java (4th Ed.). John Wiley and Sons
	C. S. Horstmann, G. Cornell (2006) Core Java 2 (J2SE 5.0) Volumen I-Fundamentos. Prentice Hall
	J. Lewis, J. Chase (2006) Estructura de datos con Java. Diseño de estructuras y algoritmos. Addison Wesley
	B. Meyer (2000) Object Oriented Software Construction. Prentice Hall
	G.J. Myers (1979) The Art of Software Testing. John Wiley & Sons
	K. Sierra, B. Bates (2005) Head First Java. OReilly
	Sun Microsystems. The Java Tutorials: Learning the Java Language
	RECURSOS WEB
EQUIPAMIENTO	Laboratorio (sala de ordenadores)
	Software: JDK SE 1.6, JUnit 4.5, Eclipse 3.5



9. Cronograma de trabajo de la asignatura

Semana	Actividades en Aula	Actividades en Laboratorio	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
1-3 (30 horas)		• Definición de Clases y Objetos. (12 horas)	• Estudio de la asignatura. (6 horas)	• Prácticas y ejercicios. (12 horas)		
4 (10 horas)		• Pruebas de Programas. (4 horas)	• Estudio de la asignatura. (2 horas)	• Prácticas y ejercicios. (4 horas)		
5 (10 horas)		• Programación modular: paquetes y visibilidad. (4 horas)	• Estudio de la asignatura. (2 horas)	• Prácticas y ejercicios. (4 horas)		
6-8 (30 horas)		• POO avanzada: herencia. (12 horas)	• Estudio de la asignatura. (6 horas)	• Prácticas y ejercicios. (12 horas)		
9 (10 horas)		• Manejo de excepciones. (4 horas)	• Estudio de la asignatura. (2 horas)	• Prácticas y ejercicios. (4 horas)		
10 (11 horas)		• POO avanzada: interfaces. (4 horas)	• Estudio de la asignatura. (2 horas)	• Prácticas y ejercicios. (4 horas)	• Examen parcial 1. (1 hora)	
11 (10 horas)		• POO avanzada: genéricos. (4 horas)	• Estudio de la asignatura. (2 horas)	• Prácticas y ejercicios. (4 horas)		
12-13 (20 horas)		• Concepto de TAD y Aplicación de los TADs para la resolución de problemas. (8 horas)	• Estudio de la asignatura. (4 horas)	• Prácticas y ejercicios. (8 horas)		
14-15 (20 horas)		• Diseño e Implementación de un TAD. (12 horas)	• Estudio de la asignatura. (4 horas)	• Prácticas y ejercicios. (6 horas)		
16 (11 horas)		• E/S en Java. (4 horas)	• Estudio de la asignatura. (2 horas)	• Prácticas y ejercicios. (4 horas)	• Examen parcial 2. (1 hora)	
TOTAL 162 horas		68 horas	32 horas	60 horas	2 horas	

Nota: Para cada actividad se especifica la dedicación en horas que implica para el alumno. Esta distribución de esfuerzos debe entenderse para el "estudiante medio", por lo que si bien puede servir de orientación, no debe tomarse en ningún caso en sentido estricto a la hora de planificar su trabajo. Cada alumno deberá hacer su propia planificación para alcanzar los resultados de aprendizaje descritos en esta Guía y ajustar dicha planificación en un proceso iterativo en función de los resultados intermedios que vaya obteniendo.



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid