



ESCUELA POLITECNICA SUPERIOR

Guía Docente

Curso 2009-2010

Titulación

Ingeniería Técnica en Informática de Sistemas

DATOS DE LA ASIGNATURA*

* Asignatura en experiencia piloto de implantación del sistema de créditos ECTS

Nombre:			
Metodología de la Programación II			
Denominación en inglés¹:			
Programming Metodology			
Código:	Año del Plan de Estudios:	Tipo:	
450004012	Publicación BOE: 27-07-2004	<input checked="" type="checkbox"/> Troncal <input type="checkbox"/> Obligatoria <input type="checkbox"/> Optativa	
Créditos:			
	Totales:	Teóricos:	Prácticos:
Créditos L.R.U.		4,50	2,25
Créditos E.C.T.S.	5,4	3,6	1,8
Departamento:			
Ingeniería Electrónica, de Sistemas Informáticos y Automática			
Área de Conocimiento:			
Ciencia de la Computación e Inteligencia Artificial			
Curso:	Cuatrimestre:	Ciclo:	
Segundo	1º Cuatrimestre	Primero	
Web de la asignatura:			
www.uhu.es/17214			

¹Para su inclusión en el Complemento Europeo al Título

DATOS DE LOS PROFESORES

Nombre:	e-mail:	Teléfono:	Despacho:
Miguel Angel Rodriguez Román	miguel.rodriguez@dti.uhu.es	959217372	65
Jesús Maria Prieto	jesus.maria@dti.uhu.es	959217641	56

DATOS ESPECÍFICOS DE LA ASIGNATURA

1.1. Descriptores de la asignatura:
Técnicas de prueba y verificación de programas. Programación orientada a objetos. Algorítmica.
1.2. Descriptores de la asignatura (en inglés)²:
Testing and programming techniques. Object oriented programming. Algorithm study.
² Para su inclusión en el Complemento Europeo al Título
2. Situación de la asignatura.
2.1. Prerrequisitos:
Conocimientos sólidos de programación estructurada en c.
2.2. Contexto dentro de la titulación:
Asignatura obligatoria de 2º de “Ingeniería Técnica en Informática de Sistemas” adscrita al Departamento de Tecnologías de la Información a en el área de Ciencias de la Computación e Inteligencia Artificial. Esta asignatura ubicada en el primer cuatrimestre de segundo curso requiere de los conocimientos adquiridos en las asignaturas de primer curso: <ul style="list-style-type: none">- Metodología de la Programación I- Estructuras de Datos I Finalmente, los conocimientos adquiridos en esta asignatura sirven de base para las asignaturas de segundo y tercer curso: <ul style="list-style-type: none">- Ingeniería del Software II- Programación Concurrente
2.3. Recomendaciones:
Se recomienda haber cursado las asignaturas de primer curso indicadas en el apartado anterior antes de comenzar esta asignatura.

3. Competencias a adquirir por los estudiantes.

3.1. Competencias transversales o genéricas.

3.1.1. Competencias instrumentales:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de análisis y síntesis.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de organización y planificación.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Comunicación oral y escrita en lengua nativa.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Conocimiento de una lengua extranjera.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de gestión de la información.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Resolución de problemas.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Toma de decisiones.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.2. Competencias personales:

<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Trabajo en equipo.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Trabajo en un equipo de carácter interdisciplinar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Trabajo en un contexto internacional.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidades en las relaciones interpersonales.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Capacidad para comunicarse con expertos de otras áreas.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Reconocimiento a la diversidad y la multiculturalidad.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Razonamiento crítico.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Compromiso ético.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.1.3. Competencias sistémicas:

<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Aprendizaje autónomo.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Adaptación a nuevas situaciones.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Capacidad de aplicar los conocimientos en la práctica.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Habilidad para trabajar de forma autónoma.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Creatividad.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Liderazgo.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Conocimiento de otras culturas y costumbres.
<input type="checkbox"/> Alto	<input checked="" type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Iniciativa y espíritu emprendedor.
<input checked="" type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Motivación por la calidad.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input checked="" type="checkbox"/> Bajo	Sensibilidad hacia temas medioambientales.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.
<input type="checkbox"/> Alto	<input type="checkbox"/> Medio	<input type="checkbox"/> Bajo	Otras: Especificar.

3.2. Competencias específicas.

3.2.1. Competencias cognitivas (saber):

Provee al alumno de conocimientos generales sobre programación orientada a objetos: clases y objetos, interfaces, reutilización y adaptación de código, colecciones de datos. Enseña a utilizar lenguajes de programación concretos de amplio uso, a transcribir a este lenguaje y ejecutar en una máquina real sus propios algoritmos.

3.2.2. Competencias procedimentales e instrumentales (saber hacer):

Enseña al alumno las técnicas básicas para la resolución de problemas mediante programas de ordenador. promueve la idea de que la construcción de programas es una metodología en la cual se deben seguir procesos sistemáticos para alcanzar el objetivo con una calidad adecuada

3.2.2. Competencias actitudinales (ser):

Acostumbra al alumno a programar respetando unas normas y directrices genéricas. Dota de capacidad analítica para enfrentarse a problemas reales y para saber elegir y aplicar las técnicas adecuadas para construir algoritmos que los resuelvan de forma eficaz y eficiente.

4. Objetivos:	
<ul style="list-style-type: none"> • Desarrollar una aproximación disciplinada a la especificación, implementación, verificación y documentación de programas. • Apreciar el papel central que juega la abstracción en la tarea de programar. • Conocer estructuras de datos, algoritmos y esquemas de uso general. • Proporcionar los fundamentos teóricos y prácticos básicos para cursar posteriores estudios en programación. • Introducir el paradigma de la programación orientada a objetos. • Aprender lenguajes de programación orientados a objetos. 	

5. Metodología (en horas de trabajo del estudiante):			
	Primer Cuatrimestre	Segundo Cuatrimestre	
	Presenciales		
Clases de teoría	32,0	0,0	
Clases de problemas	13,0	0,0	
Clases prácticas	19,5	0,0	
Actividades académicas dirigidas		0,0	
Exámenes	7,0	0,0	
	No presenciales		
Estudio de clases teóricas (factor de trabajo: 1,00)	32,0	0,0	
Estudio de clases de problemas y prácticas (factor de trabajo: 1,00)	32,5	0,0	
Preparación de actividades académicamente dirigidas y otras actividades	8,0	0,0	
Total:	144,1	0,0	
Trabajo total del estudiante: 144,1 horas.			
Horas presenciales:	64,5	Horas no presenciales:	72,6
		Exámenes:	7,0

6. Técnicas docentes.	
6.1. Técnicas docentes utilizadas:	
<input checked="" type="checkbox"/> Sesiones académicas de teoría <input checked="" type="checkbox"/> Sesiones académicas de problemas <input checked="" type="checkbox"/> Sesiones prácticas en laboratorio <input checked="" type="checkbox"/> Seminarios, exposiciones y debates <input checked="" type="checkbox"/> Trabajo en grupos reducidos <input checked="" type="checkbox"/> Resolución y entrega de problemas/prácticas <input type="checkbox"/> Realización de pruebas parciales evaluables <input type="checkbox"/> Otras: Especificar <input type="checkbox"/> Otras: Especificar	
6.2. Desarrollo y justificación:	
<i>Clases teóricas y de problemas de aula</i>	
<p>Las clases de teoría se impartirán combinando clases magistrales y problemas al total de alumnos del grupo de teoría. Durante estas sesiones el alumno adquirirá los conceptos teóricos, apoyados con ejemplos y ejercicios aclaratorios, que le capacitarán para alcanzar las competencias establecidas en el apartado 3.2 de esta guía.</p> <p>Para el desarrollo de las clases se utilizarán transparencias apoyadas con el uso de la pizarra cuando sea necesario y ordenador portátil con retroproyector para la compilación y demostración de programas en clase.</p>	

Clases de Laboratorio

Las clases prácticas se impartirán en el laboratorio utilizando un ordenador personal para cada alumno, del total de grupo de prácticas. En estas sesiones cada alumno de forma individual deberá implementar diferentes problemas utilizando Java y/o C++ como lenguaje de programación. El objetivo de estas sesiones es enfrentar al alumno con un desarrollo práctico que le permitirá adquirir las competencias procedimentales específicas, así como la mayoría de las competencias transversales.

Clases de problemas

Las sesiones de problemas pueden tener dos formatos:

En grupo :se realizaran de forma que se dividan en varios grupos que trabajaran sobre un mismo problema. En estas sesiones se planteará un problema de programación y cada grupo debe dar una solución al mismo.

Individuales: Se plantean problemas que el alumno puede entregar por escrito para que sean tenidos en cuenta su evaluación. Estos ejercicios puntuables no tienen consideración de exámenes pudiendo ser realizados sin preaviso y siempre en horario de clase, su principal finalidad es complementar las evaluaciones de aquellos alumnos de los que no se disponga evaluaciones por participación en problemas de grupo.

Exposiciones y Seminarios

Para que los alumnos conozcan aspectos avanzados de lenguajes de programación OOP se ofrecerá un seminario sobre las alternativas existentes.

7. Bloques temáticos:

Bloque 1: Propiedades Complejas de la Programación Orientada a Objetos .

- Herencia y Polimorfismo en C++.
- *JAVA vs C++.*
- *JAVA conceptos avanzados .*

Bloque 2 : Algorítmica

- Eficiencia de Algoritmos. Análisis de Algoritmos
- Algoritmos Divide y Vencerás
- Algoritmos Voraces
- Algoritmos basados en Programación Dinámica

8. Temario desarrollado:

Tema 1: Herencia y Polimorfismo en C++ y JAVA.

- 1.1 Herencia simple vs multiple. Redefinición de métodos.
- 1.2 Jerarquías de clases y visibilidad.
- 1.3 Polimorfismo y ligadura dinámica: métodos virtuales.
- 1.4 Tipos de constructores. Destructores polimorfitos. Funciones amigas.
- 1.5 Excepciones OOP.
- 1.6 Manejo de ficheros OOP.
- 1.7 Genéricos.

Tema 2: Eficiencia de Algoritmos. Análisis de Algoritmos

- 2.1 Concepto de algoritmo.
- 2.2 Introducción a la eficiencia de algoritmos.
- 2.3 El diseño de algoritmos.
- 2.4 Notaciones para la eficiencia de algoritmos.

Tema 3: Algoritmos Divide y Vencerás

- 3.1 El principio divide y vencerás.
- 3.2 Determinación del Umbral.
- 3.3 Algoritmos basados en divide y vencerás.

Tema 4: Algoritmos Voraces

- 4.1 Características de los algoritmos voraces.
- 4.2 Elementos y esquema general de un algoritmo voraz.
- 4.3 Heurísticas voraces.

Tema 5: Algoritmos basados en programación dinámica

- 4.1 Características de los algoritmos basados en programación dinámica.
- 4.2 Elementos y esquema general.
- 4.3 Resolución de problemas tipo.

9. Bibliografía.

9.1. Bibliografía general:

9.2. Bibliografía específica:

C++

Cómo Programar C++

Autores: H.M. Deitel; P.J. Deitel

Editorial: Prentice Hall

Año: 2003

Programación y Diseño en C++

Autores: J.P. Cohoon, J.W. Davidson

Editorial: McGraw--Hill

Año: 2000

Java

Piensa en Java

Autores: B. Eckel

Editorial: Prentice Hall

Año: 2002

El Lenguaje de Programación JAVA

Autores: K. Arnold, J. Gosling y D. Holmes

Editorial: Addison Wesley

Año: 2001

Estructuras de Datos en JAVA

Autores: M.Allen Weiss

Editorial: Addison Wesley

Año: 2000

Algorítmica

Algorítmica, Concepción y Análisis

Autores: G. Brassad y P.Bratley

Editorial: Masson S.A.

1.1. Específica

C++

El Lenguaje de Programación C++

Autores: B. Stroustrup

Editorial: Addison Wesley

Año: 2002

Fundamentos de Algoritmia

Editorial: Prentice Hall

Año: 1997

Estructuras de Datos y Algoritmos

Autores: A.V. Aho, J.E. Hopcroft y J. D. Ullman

Editorial: Addison Wesley.

10. Técnicas de evaluación.

10.1. Técnicas de evaluación utilizadas:

- Examen teórico-práctico
- Trabajos desarrollados durante el curso
- Participación activa en las sesiones académicas
- Controles periódicos de adquisición de conocimientos
- Examen práctico en aula de informática
- Otras: Especificar
- Otras: Especificar

10.2. Criterios de evaluación y calificación:

Examen Teórico-Práctico escrito

- El examen Teórico-Práctico consistirá en una prueba escrita compuesta por un test eliminatorio, problemas prácticos y opcionalmente preguntas cortas de teoría. La puntuación de esta prueba tendrá el 50% del total siendo obligatorio conseguir 4 puntos para hacer media con el resto de calificaciones.
- En el caso de existir test cada pregunta acertada se puntuará positivamente y cada pregunta errónea negativamente de forma proporcional al número de respuestas, no puntuando las no contestadas.
- El examen podrá incluir preguntas y/o problemas sobre los seminarios, actividades de laboratorio y actividades propuestas por el profesor durante el curso.

En caso de no superar la asignatura, la calificación de teoría no es válida en posteriores convocatorias, debiendo superar de nuevo el examen.

Para el acceso a este examen debe haberse superado con nota de apta la entrega de las prácticas de laboratorio.

Evaluación de Laboratorio

El laboratorio representa el 30% del total de la asignatura, siendo obligatorio conseguir un apto para superar la asignatura.

- Las Calificaciones de Laboratorio son validas hasta diciembre del año en curso.

Existen dos posibles itinerarios:

No presencial: Examen de laboratorio completo

- No es necesario en este itinerario la presencialidad ni la entrega de trabajos
- Esta prueba consistirá en la implementación en el lenguaje de programación Java y C++ de un problema más cuestiones prácticas de algorítmica. Esta prueba no es valorada de forma parcial, teniendo que alcanzar el alumno la calificación de 5 en el examen (de 1 a 10).

Presencial: Examen de laboratorio reducido y entrega de trabajos individuales

- Superar en defensa presencial los trabajos individuales en el laboratorio (en fecha y contenido correcto).
- Tener una asistencia mínima del 60 % a prácticas de laboratorio respetando las fechas de defensa de los trabajos obligatorios.

- La nota resultante será la resultante de la media de las prácticas entregadas más el examen reducido consistente en cuestiones a desarrollar y/o preguntas sobre la práctica desarrollada.

Evaluación de Problemas

- Los problemas desarrollados en grupos y/o ejercicios puntuables de forma individual, así como las intervenciones en las clases presenciales de teoría y práctica, permiten obtener una calificación hasta del 10% sobre el total.
- El profesor evalúa la participación de cada alumno y capacidad de reaccionar a los posibles fallos en la solución propuesta tanto en los problemas resueltos en clase como los entregados de forma escrita.
- Las calificaciones de problemas son validas hasta diciembre del año en curso.

Evaluación de Trabajos

- Existirán trabajos opcionales (Actividades Académicas Dirigidas) , entregables de forma presencial, en los que el alumno desarrolla de forma particular una práctica o un tema que quiera ampliar de los tratados en el temario.
- Dependiendo de la naturaleza del mismo la defensa del trabajo puede ser pública (presentación en clase) o sólo con el profesor.
- Las calificaciones de Trabajos son validas hasta diciembre del año en curso.

Resumen: **50% ExEscrito + 30% Laboratorio + 10% Problemas + 10% Trabajos**

*La asignatura se considera superada con una puntuación igual o superior a 5, al menos un 4 en teoría y las prácticas aptas.

11. Organización docente semanal (en horas presenciales del alumno)

11.1. Primer cuatrimestre:

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	3,0	0,0	0,0		0,0	0,0	1
2ª	3,0	0,0	0,0		0,0	0,0	1
3ª	3,0	1,0	1,5	Planificación Actividad	2,0	0,0	2
4ª	2,0	1,0	1,5		0,0	0,0	2
5ª	2,0	1,0	1,5		0,0	0,0	2
6ª	2,0	1,0	1,5	Revisión Actividad	2,0	0,0	3
7ª	2,0	1,0	1,5		0,0	0,0	3
8ª	2,0	1,0	1,5		0,0	0,0	3
9ª	2,0	1,0	1,5	Revisión Actividad	2,0	0,0	3
10ª	2,0	1,0	1,5		0,0	0,0	4
11ª	2,0	1,0	1,5		0,0	0,0	4
12ª	2,0	1,0	1,5	Entrega Actividad	2,0	0,0	4
13ª	2,0	1,0	1,5		0,0	0,0	5
14ª	1,0	2,0	1,5		0,0	0,0	5
15ª	2,0	0,0	1,5		0,0	0,0	
Periodo de exámenes						7,0	
Totales	32,0	13,0	19,5		8,0	7,0	

11.2. Segundo cuatrimestre:

Semana	Horas de clases de teoría	Horas de clases de problemas	Horas de clases prácticas	Actividades Académicas Dirigidas		Horas de exámenes	Temas del temario a tratar
				Actividad	Horas		
1ª	0,0	0,0	0,0		0,0	0,0	
2ª	0,0	0,0	0,0		0,0	0,0	
3ª	0,0	0,0	0,0		0,0	0,0	
4ª	0,0	0,0	0,0		0,0	0,0	
5ª	0,0	0,0	0,0		0,0	0,0	
6ª	0,0	0,0	0,0		0,0	0,0	
7ª	0,0	0,0	0,0		0,0	0,0	
8ª	0,0	0,0	0,0		0,0	0,0	
9ª	0,0	0,0	0,0		0,0	0,0	
10ª	0,0	0,0	0,0		0,0	0,0	
11ª	0,0	0,0	0,0		0,0	0,0	
12ª	0,0	0,0	0,0		0,0	0,0	
13ª	0,0	0,0	0,0		0,0	0,0	
14ª	0,0	0,0	0,0		0,0	0,0	
15ª	0,0	0,0	0,0		0,0	0,0	
Periodo de exámenes						0,0	
Totales	0,0	0,0	0,0		0,0	0,0	

12. Mecanismos de control y seguimiento:

- Seguimiento de actividad en clases mediante puntuación de intervenciones
- Recogida y evaluación presencial de trabajos opcionales

- Recogida y defensa presencial de prácticas de laboratorio.
- Firma de hoja de asistencia a clases laboratorio.