

Metodología de la Programación

GUÍA DOCENTE

Curso 2010-2011

Titulación:	Grado en Matemáticas			701G	
Asignatura:	Metodología de la Programación			701104004	
Materia:					
Módulo:	M2 Informática				
Carácter:	Básica	Curso:	1º	Semestre:	1º
Créditos ECTS:	6	Horas presenciales:	60	Horas de trabajo autónomo estimadas:	90
Idiomas en los que se imparte:	español				
Idiomas del material de lectura o audiovisual:	español				

Departamentos responsables de la docencia:

Matemáticas y Computación				Código	
Dirección:	Luis de Ulloa, s/n	Código postal:	26006		
Teléfono:	+34 941 299 452	Fax:	+34 941 299 460	Correo electrónico:	dmc@unirioja.es
				Código	
Dirección:				Código postal:	
Teléfono:	+34 941 299	Fax:	+34 941 299	Correo electrónico:	@unirioja.es

Profesores

Profesor responsable de la asignatura:	Juan Félix San Juan Díaz			
Teléfono:	+34 941 299 440	Correo electrónico:	juanfelix.sanjuan@dmc.unirioja.es	
Despacho:	230	Edificio:	Juan Luis Vives	
Horario de tutorías:				
Nombre profesor:	<i>(Campos a completar al inicio de curso)</i>			
Teléfono:	+34 941 299	Correo electrónico:	@unirioja.es	
Despacho:		Edificio:		
Horario de tutorías:				
Nombre profesor:				
Teléfono:		Correo electrónico:		
Despacho:		Edificio:		
Horario de tutorías:				

Descripción de contenidos:

- Introducción a la Programación.
 - Elementos de un programa: datos y acciones.
 - Nociones básicas sobre lenguajes de programación.
 - Algoritmo. Estrategias para el diseño de algoritmos.
- Programación imperativa.

- Tipos de datos y operadores.
- Sentencias.
- Subalgoritmos. Comunicación entre subalgoritmos.
- Mecanismos básicos para la construcción de estructuras de datos.
- Nociones sobre análisis de la eficiencia de algoritmos.

Requisitos previos:

PROGRAMA GENERAL

Contexto:

La asignatura de *Metodología de la programación* introduce a los alumnos en los conocimientos básicos que le serán necesarios para poder comprender y desarrollar aplicaciones software en el futuro.

Competencias:

Competencias generales: CG7, CG8, CG9.

- **CG 7.** Saber abstraer las propiedades estructurales de objetos de la realidad observada y de otros ámbitos, distinguiéndolas de aquellas puramente ocasionales, comprobando la aplicabilidad de las Matemáticas.
- **CG 8.** Capacitar para el aprendizaje autónomo de nuevos conocimientos y técnicas.
- **CG 9.** Capacitar para el trabajo en equipo, comprendiendo el contexto matemático o interdisciplinar en que se realiza.

Competencias específicas: CE1, CE4, CE5.

- **CE 1.** Resolver problemas de Matemáticas, mediante habilidades de cálculo básico y otras técnicas, planificando su resolución en función de las herramientas de que se disponga y de las restricciones de tiempo y recursos.
- **CE 4.** Encontrar soluciones algorítmicas de problemas matemáticos y de aplicación (**de ámbito académico, técnico, financiero o social**), **sabiendo comparar distintas alternativas, según criterios de adecuación, complejidad y coste.**
- **CE 5.** Saber programar algoritmos de modo correcto y eficaz, eligiendo convenientemente lenguajes y plataformas de programación.

Resultados del aprendizaje:

- Conocer los conceptos básicos de la programación imperativa.
- Conocer la sintaxis de algún lenguaje de programación imperativa (preferentemente que soporte un posterior enfoque orientado a objetos).
- Aprender metodología/estrategias para la resolución de problemas que lleven a soluciones algorítmicas bien diseñadas.
- Desarrollar programas de tamaño pequeño/mediano.
- Aprender a usar los mecanismos de construcción de estructuras de datos para la representación y manejo de información.
- Aprender a realizar una primera estimación de la complejidad en tiempo en casos sencillos. Aplicación sobre algunos algoritmos clásicos.

Temario:

Tema 1 INTRODUCCIÓN A LA PROGRAMACIÓN DE ORDENADORES

- 1.1. ¿Qué es la informática y la programación?
- 1.2. Componentes fundamentales de la informática.

Tema 2 PROBLEMAS, ALGORITMOS Y PROGRAMAS

- 2.1. Fases en la obtención de un programa.
- 2.2. Definiciones e ideas que aproximan la noción de algoritmo.
- 2.3. Especificación de algoritmos.
- 2.4. Estructura de un algoritmo.
- 2.5. Diseño por refinamientos sucesivos. Análisis descendente.

Tema 3 TIPOS DE DATOS SIMPLES Y SENTENCIAS ELEMENTALES

- 3.1. Noción de tipo de dato.
- 3.2. Clasificación de los tipos de datos.
- 3.3. Variables, constantes y expresiones.
- 3.4. Sentencias elementales.
- 3.5. Traducción de pseudocódigo a C++.

Tema 4 ESTRUCTURAS DE CONTROL

- 4.1. Estructura secuencial.
- 4.2. Estructura condicional.
- 4.3. Estructura iterativa.

Tema 5 DISEÑO DESCENDENTE DE ALGORITMOS

- 5.1. Introducción.
- 5.2. Funciones.
- 5.3. Acciones.
- 5.4. Clases de parámetros formales.

Tema 6 TIPOS DE DATOS ESTRUCTURADOS

- 6.1. Registros.
- 6.2. Vectores.

Bibliografía:

Para la parte teórica:

- Joyanes L., Zahonero I., "Algoritmos y estructuras de datos: una perspectiva en C", Mc Graw-Hill, 2004.
La parte I del libro recoge la práctica totalidad del programa de la asignatura. El nivel con el que se tratan los temas es básico.
- Castro J., Cucker F., Meseguer X., Rubio A., Solano A., "Curso de Programación", Ed. McGraw-Hill, 1993.
Buen libro de introducción y se adapta aceptablemente a los contenidos de la asignatura.
- Joyanes L., Rodríguez L., Fernández M., "Fundamentos de programación: libro de problemas", Ed. McGraw-Hill, 2ª ed., 2003.
La parte I recoge la práctica totalidad del programa de la asignatura.
- Joyanes L., "Problemas de metodología de la programación", McGraw-Hill, 1990.
Los nueve primeros capítulos recogen la práctica totalidad del programa de la asignatura.

Para la parte práctica en C++:

- Harvey M., "Cómo programar en C, C++ y Java", Pearson Educación, 4ª ed., 2004.
- Ellis M., Stroustrup B., "Manual de referencia C++ con anotaciones", Addison Wesley/Díaz de Santos, 1994.
- Schildt H., "Lenguaje C++: guía de autoenseñanza", McGraw-Hill, 2ª Ed., 1995.

Tutoriales del entorno Dev-C++: <http://www.bloodshed.net/dev/doc/index.html>

Metodología:

Modalidades organizativas:

Métodos de enseñanza:

- MO1: Clases teóricas	- ME1: Lección magistral
- MO3: Clases prácticas	- ME3: Resolución de ejercicios y problemas
- MO5: Tutorías	- ME4: Utilización de recursos informáticos
- MO6: Estudio y trabajo autónomo del alumno	

Organización:

Actividades presenciales:	Horas
Clases teórico-prácticas donde se desarrollan los contenidos	32
Clases prácticas en aula informática	28

Total horas presenciales 60

Actividades no presenciales (trabajo autónomo):	Horas estimadas
Estudio autónomo individual o en grupo	25
Resolución individual de ejercicios, cuestiones u otros trabajos, actividades en biblioteca o similar	40
Preparación de las prácticas y elaboración de cuaderno de prácticas	25

Total horas estimadas de trabajo autónomo 90

Total horas 150

Evaluación

Sistemas de evaluación:	% sobre total	Recuperable/ No Rec.
Prueba teórico-práctica en el aula de informática	10%	Rec.
Prueba teórico-práctica	10%	No Rec.
Trabajo final de curso	10%	Rec.
Examen al final del semestre	70%	Rec.

Crucios críticos para superar la asignatura:

Para poder promediar será necesario sacar al menos un 5 en el examen final.