

# ORGANIZACIÓN DOCENTE del curso 2008-09

## 1. DATOS GENERALES DE LA ASIGNATURA

|                             |  |                               |                          |   |         |
|-----------------------------|--|-------------------------------|--------------------------|---|---------|
| <b>NOMBRE</b>               | Programación 1   |                               | <b>PÁGINA WEB</b>        | www.ctr.unican.es/asignaturas/programacion1 |         |
| <b>CÓDIGO</b>               |  |                               |                          |   |         |
| <b>DEPARTAMENTO</b>         | Matemáticas, Estadística y Computación                     |                               |                          |   |         |
| <b>PLAN DE ESTUDIOS</b>     | Ingeniero en Informática                                   |                               | <b>CURSO</b>             | 2008/2009                                   |         |
| <b>PROFESORADO</b>          | <b><u>Nombre</u></b>                                       |                               | <b><u>e-mail</u></b>     |   |         |
|                             | Michael González Harbour                                   |                               |                          |   |         |
|                             | Profesor Ayudante Doctor (a contratar)                     |                               |                          |   |         |
|                             | Profesor asociad (a contratar)                             |                               |                          |   |         |
|                             |  |                               |                          |   |         |
|                             |  |                               |                          |   |         |
|                             |  |                               |                          |   |         |
|                             |  |                               |                          |   |         |
| <b>CRÉDITOS ALUMNO</b>      | <u>Teóricos</u><br>(1)                                     | <u>Prac. Problemas</u><br>(2) | <u>Prac. Laboratorio</u> | <u>Prac. Computador</u>                     | TOTALES |
|                             | 4.5  | 1.5                           |                          | 3   | 9       |
| <b>LUGAR DE IMPARTICIÓN</b> | <u>Teóricos</u>  | <u>Prac. Problemas</u>        | <u>Prac. Laboratorio</u> | <u>Prac. Computador</u>                     |         |
|                             |  |                               |                          |   |         |
| <b>HORARIO PREVISTO(*)</b>  | <u>Teóricos</u>  | <u>Prac. Problemas</u>        | <u>Prac. Laboratorio</u> | <u>Prac. Computador</u>                     |         |
|                             |  |                               |                          |   |         |
| <b>(*) Observaciones:</b>   | Consultar el cuadro que se expone en el tablón de anuncios |                               |                          |   |         |

(1) Se corresponde con clases magistrales de teoría en aula

(2) Se corresponde con clases prácticas (problemas, experiencias de cátedra,...) en aula

## **2. PROGRAMA DE LA ASIGNATURA**

### **Bloque I: Conceptos y técnicas básicas en programación**

#### 1. Introducción.

- Programas y lenguajes.
- Necesidad de la programación modular.
- Ciclo de vida de los programas
- Concepto de algoritmo
- Noción de proceso.
- Variables y estado de un programa
- Traza de un proceso

#### 2. Datos y expresiones. Especificación de algoritmos

- Tipos primitivos
- Variables y constantes
- Operadores y expresiones
- Noción de especificación
- Noción de predicado
- Corrección de un algoritmo
- Reglas de consecuencia de una especificación
- Encapsulamiento: Concepto de clase y objeto
- Diagramas de clases
- Métodos y paso de parámetros
- Especificación de clases y objetos

#### 3. Estructuras algorítmicas básicas.

- La asignación
- Corrección de la asignación. Regla de inferencia
- Composición secuencial de instrucciones
- La instrucción alternativa simple
- La instrucción alternativa múltiple
- Alternativa exclusiva y general

#### 4. Iteración y recursión.

- Diseño iterativo. Instrucción de bucle
- Corrección de la implementación iterativa: Trazas, invariantes y cotas
- Fases del diseño iterativo
- Otras instrucciones de bucle
- Recursión
- Corrección de la implementación recursiva
- Fases del diseño recursivo

#### 5. Iteración y recursión sobre secuencias.

- Introducción
- Descripción de la secuencia. Interfaz
- Recorridos sobre secuencias
- Búsquedas en secuencias
- Esquemas mixtos

## 6. Iteración y recursión sobre tablas.

- Concepto de tabla. Sintaxis
- Operaciones sobre tablas
- Recorrido de tablas
- Búsqueda en tablas
- Búsqueda binaria
- Algoritmos sencillos de ordenación en tablas

## **Bloque II: Elementos del lenguaje de programación Java**

### 1. Introducción a los lenguajes de programación

- El software del sistema.
- Lenguajes de Programación.
- Lenguajes de alto y bajo nivel.
- Ensambladores, compiladores e intérpretes.
- Algunos lenguajes de alto nivel.
- El proceso de compilación.
- El ciclo de vida del software

### 2. Estructura de un programa

- Introducción al lenguaje Java
- Clases y objetos en Java
- Estructura de un programa
- Estructura de una clase
- Estructura de una operación

### 3. Datos y expresiones simples

- Tipos primitivos
- Los datos en Java
- Operadores y expresiones
- Conversión de tipos
- Uso de funciones matemáticas
- Declaración de clases y objetos
- Strings
- Composición de objetos

### 4. Instrucciones de control.

- Instrucciones simples y compuestas
- Instrucción condicional simple
- Instrucción condicional múltiple
- Instrucciones de bucle
- Bucle con condición de permanencia al principio
- Bucle con condición de permanencia al final
- Bucle con variable de control
- Instrucciones de salto en bucles

### 5. Entrada/Salida simple.

- Entrada salida de texto
- Entrada/salida de números
- Dibujos
- Gráficas de funciones
- Menús

## 6. Arrays, secuencias y tablas

- Construcción de tablas mediante arrays
- Arrays unidimensionales
- Uso de arrays
- Recorrido y recorrido parcial
- Búsqueda
- Arrays multidimensionales
- Tablas de tamaño variable

### **Bloque III: Herramientas de laboratorio**

#### 1. Sistemas operativos

- Conceptos básicos
- Sistemas operativos comunes
- Inicio y terminación de sesiones
- El sistema de ficheros
- Nombres de ficheros en Unix
- Directorio de trabajo
- Intérprete de órdenes
- Órdenes más usuales
- Gestión de directorios
- Gestor gráfico de ficheros
- Uso de memorias USB
- Aspectos avanzados

#### 2. Entorno integrado de desarrollo de programas

- Proceso de desarrollo de programas
- El editor de texto
- Compilar y ejecutar programas Java
- Uso del entorno de desarrollo
- Gestión de proyectos
- La compilación
- La depuración
- Generación de documentación
- Empaquetamiento del programa

### **Asignaturas que se recomienda al alumno haber cursado o estar cursando**

Ninguna

### **3. OBJETIVOS GENERALES DE LA ASIGNATURA**

Aprender los conceptos básicos de algorítmica, así como la codificación de algoritmos sencillos en un lenguaje de programación concreto.

#### 4. OBJETIVOS ESPECIFICOS: APTITUDES/DESTREZAS

##### Conocimientos

- Conocer y comprender la sintaxis y semántica de las expresiones e instrucciones de un lenguaje de programación imperativo.
- Conocer el concepto de clase y objeto como elementos constituyentes de los programas
- Conocer y saber utilizar los tipos de datos elementales, las tablas y las secuencias.
- Conocer los conceptos de método y paso de parámetros.
- Conocer los principios de la entrada/salida interactiva simple

##### Habilidades

- Especificar problemas sencillos de programación
- Saber aplicar a problemas sencillos el razonamiento inductivo para construir algoritmos.
- Codificar estos algoritmos en un lenguaje de programación imperativo
- Utilizar un sistema de desarrollo para editar, compilar y ejecutar programas
- Razonar informalmente sobre la corrección de los algoritmos

##### Capacidades

- Dotar de capacidad para la abstracción (simplificación) y para el razonamiento semiformal pero riguroso.
- Capacidad para comprender los problemas, desde el enunciado, saber especificar y distinguir datos, resultados y datos/resultados.
- Capacidad para enfrentarse a los problemas, siendo capaz de resolver nuevas situaciones desde los conocimientos adquiridos.
- Dotar de capacidad para abordar con éxito el estudio de nuevas materias de la titulación de Ingeniería Informática.

#### 5. BIBLIOGRAFÍA

##### Básica

- Castro J., et al. "Curso de Programación", McGrawHill, 2001.
- 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

### **Complementaria**

- Ken Arnold, James Gosling, David Holmes, "El lenguaje de programación Java", 3ª edición. Addison-Wesley, 2000, ISBN 84-7829-045-1
- Shen A. "Algorithms and Programming: problems and solutions", Birkhauser, 1997.
- E. Bueno, y otros. "Algoritmos y Ejercicios resueltos en Java". Prentice Hall 2003. ISBN 84-705-4024-2
- B. Eckel, "Piensa en Java", Addison Wesley, 2002. ISBN 84-205-3192-8
- Bishop J. "Java. Fundamentos de programación", Addison Wesley, 1999.

## **6. ACTIVIDADES A DESARROLLAR EN LA ASIGNATURA**

Se desarrollarán una serie de prácticas de laboratorio en relación con los contenidos de la asignatura.

Cada práctica se desarrollará en tres fases:

- La primera consiste en la presentación por parte del profesor a través de material escrito y explicaciones generales de la traslación al lenguaje de programación de los conceptos estudiados en teoría.
- La segunda fase consiste en la resolución de uno o varios ejercicios por parte del alumno para los que dispone de un tiempo a determinar en cada caso. La resolución de cada ejercicio implica las fases habituales en la construcción de programas: especificación del problema, diseño en pseudocódigo, codificación en el lenguaje de programación, y prueba del programa o módulos desarrollados.
- Adicionalmente se suministrará una colección de ejercicios de la asignatura relacionados con las fases de especificación y diseño en pseudocódigo. Algunos de ellos serán resueltos por el profesor para ilustrar la aplicación de los contenidos suministrados en las clases magistrales, otros seleccionados deberán ser resueltos por los alumnos.

## **7. MÉTODO DE EVALUACIÓN**

PROCEDIMIENTO ORDINARIO: evaluación mixta, con un 40% de peso correspondiente al trabajo continuo en las prácticas de la asignatura y un 60% al examen final de cuestiones y problemas.

**PROCEDIMIENTO EXTRAORDINARIO:** se podrá admitir en casos excepcionales y justificados una evaluación única de las prácticas en el laboratorio, mediante un examen de prácticas. A este examen se podrá optar tras la entrega de los informes de las prácticas y ejercicios obligatorios de la asignatura. El examen final de cuestiones y problemas es el mismo que el del procedimiento ordinario. Los pesos de cada examen también (40% examen de prácticas, y 60% examen de cuestiones y problemas).

En ambos procedimientos es preciso superar por separado las prácticas y el examen de cuestiones y problemas.

**Descripción de la evaluación continua: actividades que debe desarrollar el alumno y su valoración**

La evaluación continua se realizará a través de la valoración de las prácticas y de la colección de ejercicios que deben ser resueltos por el alumno.

Criterios de evaluación de las prácticas:

a) Trabajo en el laboratorio

- Conocimientos: Conocimiento de los conceptos de algoritmos y del lenguaje Java
- Grado de Resolución: Grado de resolución de la práctica alcanzado durante las horas en el laboratorio.

b) Informes sobre las prácticas

- Especificación y diseño: Facilidad para especificar diseñar y comprender algoritmos.
- Estilo: Estilo de programación, claridad del código, documentación del código.
- Organización: Organización del informe, claridad en la exposición
- Fecha de entrega: Entrega del informe dentro del plazo marcado, con el objetivo de conseguir una evaluación realmente continuada.

El profesor de prácticas puede pedir que se muestre el correcto funcionamiento de la práctica en el laboratorio. Las prácticas cuyo informe se presente con retraso de hasta una semana tendrán un punto menos (sobre 10) y las de informes presentados con retraso entre una y dos semanas tendrán tres puntos menos. Las entregas pasadas dos semanas del plazo no se evaluarán, pero su presentación es obligatoria.

**Descripción del examen final (duración, se pueden llevar apuntes o no, tiene partes diferenciadas o no, se promedian teoría y problemas o no, etc).**

El examen final escrito de cuestiones y problemas constará de dos partes. La primera parte se referirá a la evaluación de conocimientos teóricos para que lo que se plantearán una serie de preguntas y ejercicios de respuesta rápida. La segunda parte se referirá a la comprensión y utilización del lenguaje de programación para lo que se planteará un problema de dificultad media. Ambas partes tienen la misma puntuación global y tendrán un tiempo asignado de 90 minutos. Se podrán utilizar apuntes y bibliografía.

El examen de prácticas del procedimiento extraordinario es un examen en el laboratorio con una duración de dos horas en las que se pedirá el diseño de un algoritmo especificado por el profesor y su codificación en lenguaje de programación, utilizando los equipos del laboratorio.

## **8. OBSERVACIONES**

Para superar la asignatura es preciso superar tanto el examen como las prácticas. La nota final estará compuesta por:

- 60% examen escrito de cuestiones y problemas
- 40% nota de prácticas

En caso de que una de estas partes no se supere, la nota final será el máximo de 4.5 y la media obtenida.

En caso de aprobar únicamente una de las dos partes en la convocatoria de junio, se guardaría la nota de esa parte para septiembre.