UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

GUÍA DOCENTE *CURSO: 2010/11*

12688 - TECNOLOGÍA DE LA PROGRAMACIÓN

ASIGNATURA: *12688 - TECNOLOGÍA DE LA PROGRAMACIÓN*
CENTRO: *Escuela de Ingeniería Informática*
TITULACIÓN: *Ingeniero en Informática*
DEPARTAMENTO: *INFORMÁTICA Y SISTEMAS*
ÁREA: *Lenguajes Y Sistemas Informáticos*
PLAN: *10 - Año 199* **ESPECIALIDAD:**
CURSO: *Segundo curso* **IMPARTIDA:** *Primer semestre* **TIPO:** *Troncal*
CRÉDITOS: *7,5* **TEÓRICOS:** *4,5* **PRÁCTICOS:** *3*

Información ECTS

Créditos ECTS: No

Descriptores B.O.E.

Diseño de Algoritmos. Análisis de Algoritmos. Técnicas de Verificación y Pruebas de Programas

Temario

MÓDULO 0: LENGUAJE C (12 horas, 12T)
Tema 0: Lenguaje C
Bibliografía básica: [DI98]
Bibliografía complementaria: [BA91]

MÓDULO 1: INTRODUCCIÓN A LOS CONCEPTOS DE INGENIERÍA DEL SOFTWARE (1 hora, 1T)
Tema 1: Introducción a la ingeniería del software
Bibliografía básica: [DI04]
Bibliografía complementaria: [PR93]

MÓDULO 2: VERIFICACIÓN FORMAL DE ALGORITMOS (16 horas, 7T+9P)
Tema 2: Introducción a la verificación. Lógica de predicados
Tema 3: Verificación de algoritmos iterativos
Tema 4: Verificación de algoritmos recursivos
Tema 5: Derivación de algoritmos
Tema 6: Prueba de programas
Bibliografía básica: [DI04]
Bibliografía complementaria: [PE97]

MÓDULO 3: ANÁLISIS DE LA EFICIENCIA DE ALGORITMOS (16 horas, 7T+9P)
Tema 7: Introducción al análisis de la eficiencia de los algoritmos
Tema 8: Notaciones asintóticas
Tema 9: Análisis de la eficiencia de algoritmos iterativos
Tema 10: Resolución de recurrencias

Tema 11: Análisis de la eficiencia de algoritmos recursivos
Bibliografía básica: [DI04]
Bibliografía complementaria: [BR98]

MÓDULO 4: DISEÑO DE ALGORITMOS (14 horas, 7T+7P)
Tema 12: Algoritmos voraces
Tema 13: Divide y vencerás
Tema 14: Programación dinámica
Tema 15: Vuelta atrás
Bibliografía básica: [DI04]
Bibliografía complementaria: [BR98], [CO89]

MÓDULO 5: COMPLEJIDAD COMPUTACIONAL (1 hora, 1T)
Tema 16: Introducción a los problemas NP-Completos
Bibliografía básica: [DI04]
Bibliografía complementaria: [BR98], [CO89]

Requisitos Previos

Que el alumno haya cursado con éxito:
- Metodología de la Programación
- Estructuras de Datos I

También es importante tener conocimientos de inglés que permitan la lectura de documentos técnicos.

Objetivos

- Adquirir destreza en el uso del lenguaje de programación C
- Verificar la correctitud de algoritmos
- Probar y depurar programas
- Analizar la eficiencia de algoritmos
- Utilizar estrategias conocidas para diseñar algoritmos
- Aplicar los conocimientos adquiridos a la resolución de problemas

Metodología

La metodología que se va a utilizar está orientada a los estilos de aprendizaje de los alumnos, así se contempla en el Proyecto metodológico que se va a desarrollar en esta asignatura y que fue aprobado al grupo TILDE por el Vicerrectorado de Calidad e Innovación Educativa. Para ello, es necesario e imprescindible conocer la forma de aprender de cada alumno con apreciación de la existencia de diferentes estilos de aprendizaje, permitiendo una atención individualizada por perfil que motive al estudiante, le aporte confianza en sus capacidades y potencie su autoestima y autonomía. Y además, que aproxime y mejore la relación enseñanza-aprendizaje.

Sistema de apoyo metodológico:

1. La dotación de medios en el aula es esencial para aplicar la metodología propuesta y poder usar la tecnología para buscar información, consultar fuentes y disponer de recursos didácticos de forma inmediata. Se requieren mesas colocadas en círculo, con capacidad para un número de alumnos entre siete y diez, y con un portátil o PC fijo para cada estudiante, además de una pizarra por equipo.

2. La composición de los equipos se establece teniendo en cuenta diferentes estilos de aprendizaje,

que aporten la diversidad requerida para el trabajo colaborativo haciendo que los resultados de aprendizaje obtenidos sean más enriquecedores y completos, debido a la interacción y al diálogo entre estudiantes con perfiles distintos. Es una forma de incentivar la colaboración en una colectividad no competitiva y fortalecer el sentimiento de solidaridad y respeto mutuo, a la vez que disminuye la sensación de aislamiento.

Los quehaceres diarios en el aula se resumen a continuación:

1. En el horario establecido por el centro para las clases en el aula, se utilizan dos modalidades para los contextos de las actividades que la profesora y los alumnos desarrollarán en el proyecto: a nivel de grupo y a nivel de equipo.

2. La profesora presenta a todo el grupo una síntesis del contenido de cada tema, publica el material disponible en la página web de la asignatura, indica las tareas individuales y colectivas a realizar y establece un período adecuado para la adquisición de competencias y habilidades específicas.

3. Este lapso de tiempo se consume en las sesiones de trabajo colaborativo, cada equipo planifica y organiza su labor, que incluye la ejecución de las tareas colectivas con discusión de aportaciones individuales y resolución de dudas entre iguales; además, debido a la presencia de la profesora en el aula comprende la realimentación inmediata.

4. El tiempo que dedica la profesora a cada equipo durante las horas lectivas será aproximadamente el mismo, intentando optimizar su presencia en cada uno para poder ocuparse de todos los equipos. Los quehaceres que realiza son la resolución de dudas, la valoración del grado de participación y de las aportaciones, tanto individuales como colectivas, y la evaluación del progreso de cada estudiante.

En las clases del laboratorio los alumnos realizan ejercicios de programación en Lenguaje C y la profesora atiende individualmente a cada uno. La realización de las clases prácticas se soporta íntegramente vía Web.

El medio principal para suministrar información referente a la asignatura es la página Web http://aulaga.dis.ulpgc.es.

Criterios de Evaluación

La evaluación de la asignatura consta de dos partes: una práctica y otra teórica. En la parte práctica se recogen las prácticas de laboratorio, un examen escrito práctico y un examen práctico de rescate. En la parte teórica se realiza evaluación continua y examen escrito final.

Evaluación de la parte teórica
* Evaluación continua

Se lleva a cabo por medio de cinco exámenes escritos que constan de ejercicios relativos a los temas tratados. Cada examen aporta el mismo porcentaje a la nota de esta parte, un 10%, y se valora de 0 a 10 puntos. Además, cada alumno realiza una coevaluación de sus compañeros de equipo por medio de un cuestionario, se hace la media de los resultados obtenidos en tres ocasiones a lo largo del semestre. El porcentaje que aporta la coevaluación a la nota de esta parte es del 10%. Requiere la asistencia a clase debido a la metodología orientada a los estilos de aprendizaje y a la coevaluación.

* Examen escrito final

El examen escrito final está dividido en ejercicios, cada uno de los cuales aporta un porcentaje a la nota del examen que se especifica en el enunciado. Este examen se valora de 0 a 10 puntos.

Evaluación de la parte práctica
* Prácticas de laboratorio

La evaluación de las prácticas se efectúa a partir de los ejercicios prácticos que se van planteando regularmente. El enunciado de cada ejercicio incluye las especificaciones, plazos de entrega y porcentaje que aporta a la evaluación.

Por cada ejercicio práctico planteado el alumno debe entregar, dentro de los plazos establecidos, los ficheros conteniendo el código fuente y demás información requerida, usando el medio que se establezca.

El profesor somete los programas e información entregados a los análisis y pruebas oportunos para formular una valoración global que tenga en cuenta todos los aspectos implicados en su realización, con especial énfasis en la ejecución (funcionamiento, adecuación a las especificaciones, robustez, ...) y en el estilo (formato, comentarios, elección de identificadores, ...).

Cuando lo estime conveniente, el profesor puede citar al alumno para formularle cuestiones que considere relevantes para la valoración global reseñada. Las prácticas en el laboratorio se valoran de 0 a 10 puntos.

* Examen escrito práctico

Examen tipo test de lenguaje C que se valora de 0 a 10 puntos.

* Examen práctico de rescate

Este examen consiste en la realización de un ejercicio de programación en una situación real ante el ordenador. Su finalidad es valorar la efectividad de las habilidades alcanzadas en este ámbito.

Se valora con los mismos criterios que las prácticas pero se califica como APTO o NO APTO.

Calificación

La valoración de la parte práctica consta, a partes iguales, de las prácticas de laboratorio y del “examen escrito práctico”. Además, para superar esta parte se requiere obtener la calificación de APTO en el examen práctico de rescate. De este último examen quedan exonerados quienes obtengan resultados especialmente brillantes en el examen escrito práctico y/o en la valoración de las prácticas de laboratorio. No pueden ser exonerados de este examen los alumnos que no asistan al menos al 60% de las clases prácticas.

Para aprobar la asignatura hay que superar la parte práctica y la teórica, además de obtener una nota mínima de 5 puntos en la nota final. La nota de la parte teórica se puede obtener a través de la evaluación continua o bien mediante el examen escrito final, si no se ha superado esta parte con el sistema previo. La nota final se calcula con la siguiente fórmula:

NF = NT * 0.6 + NP * 0.4

donde NF representa la nota final, NT la nota de la parte teórica, NP la nota de la parte práctica. La nota final de los alumnos que no cumplan las condiciones estipuladas para superar la asignatura es

la obtenida en la parte teórica hasta un máximo de 4 puntos.

Los alumnos que incurran en fraude en alguna de las pruebas obtienen una nota final de 0 puntos, independientemente de otras medidas que pudieran tomarse de acuerdo con lo establecido en los reglamentos, al respecto, de la ULPGC.

Descripción de las Prácticas

Las prácticas consistirán en el desarrollo de algoritmos para solucionar problemas simples y la implementación de pequeños programas o módulos de programas que utilicen los diferentes conceptos y elementos de programación presentados en las clases teóricas. En las prácticas en laboratorio, los programas se implementarán usando el lenguaje de programación C y deberán funcionar en el ordenador como requisito previo a cualquier evaluación.
Las prácticas incluirán tanto ejercicios de acción formativa como de acción sumativa que se señalarán adecuadamente en cada caso y para los que se establecerán plazos de entrega que deberán ser respetados para que sean tenidos en cuenta a efectos de evaluación.
Las prácticas de acción sumativa son:

Práctica número 1
Descripción: Entorno de Programación. Estructura básica de un programa.
Objetivos:
Familiarizar al alumno con el entorno de programación y que inicie su experiencia en el lenguaje desarrollando un programa simple.
Nº horas estimadas en laboratorio: 2 horas

Práctica número 2
Descripción: Tipos de datos no homogéneos.
Objetivos:
Adquirir destreza en la manipulación de estructuras de datos complejas utilizando programas compuestos de varios ficheros
Nº horas estimadas en laboratorio: 4 horas

Práctica número 3
Descripción: Ristras y ficheros.
Objetivos:
Adquirir destreza en la manipulación ristras de caracteres y de ficheros en C.
Nº horas estimadas en laboratorio: 4 horas

Práctica número 4
Descripción: Comparación de resultados empíricos de dos métodos de ordenación.
Objetivos:
Comparación de la eficiencia teórica de dos métodos de ordenación con los resultados empíricos obtenidos en el laboratorio.
Nº horas estimadas en laboratorio: 5 horas

Material de Laboratorio requerido.
Software: Entorno de desarrollo de C estándar que incluya editor, compilador y depurador. Navegador de internet actualizado.
Hardware: Un ordenador por alumno que ejecute el software solicitado y que disponga de acceso a la red de la universidad.

Bibliografía

[1 Básico] Metodología y tecnología de la programación II /

Margarita Díaz Roca, Juan Carlos Rodríguez del Pino.

Universidad de Las Palmas de Gran Canaria, Vicerrectorado de Planificación y Calidad,, Las Palmas de Gran Canaria : (2004)

84-96131-95-5

[2 Básico] Introducción al lenguaje de programación C /

Margarita Díaz Roca, Juan Carlos Rodríguez del Pino, Zenón Hernández Figueroa.

Autor-editor,, Las Palmas : (1998)

848416862X

[3 Recomendado] El Lenguaje de programación en C :diseño e implementación de programas /

Félix García Carballeira [et al.].

Pearson Education,, Madrid : (2002)

8420531782

[4 Recomendado] Fundamentos de algoritmia /

G. Brassard, P. Bratley.

Prentice Hall,, Madrid : (1998)

848966000X

[5 Recomendado] The C book: featuring the ANSI C Standard /

Mike Banahan, Declan Brady, Mark Doran.

Addison-Wesley,, Wokingham (England) : (1991) - (2nd ed.)

0-201-54433-4

[6 Recomendado] Diseño de programas. Formalismo y abstracción /

Ricardo Peña Marí.

..T250:

Prentice Hall,, MadridMadrid : (2004)

84-205-4191-5

[7 Recomendado] Ingeniería del software: un enfoque práctico /

Roger S. Pressman.

McGraw-Hill,, México : (2006) - (6ª ed.)

970-10-5473-3

Equipo Docente

MARGARITA DÍAZ ROCA *(COORDINADOR)*

Categoría: CATEDRATICO DE ESCUELA UNIVERSITARIA

Departamento: INFORMÁTICA Y SISTEMAS

Teléfono: 928458732 **Correo Electrónico:** margarita.diaz@ulpgc.es

WEB Personal: http://www2.dis.ulpgc.es/~mdiaz/

Resumen en Inglés

The purpose of the Tecnología de la Programación course is to provide appropriate methods and techniques for building quality software. It focuses on two of the factors that define quality: correctness and efficiency. The basic factor is the consistency between the algorithm and the specification, therefore, we begin with the formal verification as a tool for demonstrating the correctness. The efficiency analysis is a good criterion to distinguish between different solutions to a problem or improve a proposal. We also study a set of fundamental techniques for designing

algorithms that provides a sufficient background to deal effectively with problem-solving.