



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

GUÍA DOCENTE

CURSO: 2007/08

14066 - PROGRAMACIÓN

ASIGNATURA: 14066 - PROGRAMACIÓN

CENTRO: Escuela de Ingeniería de Telecomunicación y Electrónica

TITULACIÓN: Ingeniero de Telecomunicación

DEPARTAMENTO: INGENIERÍA TELEMÁTICA

ÁREA: Ingeniería Telemática

PLAN: 13 - Año 200**ESPECIALIDAD:**

CURSO: Primer curso **IMPARTIDA:** Segundo semestre **TIPO:** Obligatoria

CRÉDITOS: 9

TEÓRICOS: 6

PRÁCTICOS: 3

Descriptores B.O.E.

Ordenación. Recursividad. Estructuras de datos: listas, árboles y grafos. Metodología de programación. Desarrollo e implementación de programas. Estructuras de datos, arrays y archivos.

Temario

TEMA 1: Arrays [4+2 horas]

- Array No Ordenado: Interfaz y Cuerpo
- Array Ordenado. Búsqueda Binaria
- Verificación: Test
- Análisis: Orden del algoritmo

TEMA 2: Programación orientada a objeto. [4+2 horas]

- Definición de objetos.
- Herencia.
- Ligadura dinámica.
- Interfaces.

TEMA 3: Métodos Simples de Ordenación [4+2 horas]

- Burbuja
- Selección
- Inserción

TEMA 4: Listas Enlazadas [6+4 horas]

- Nodos, enlaces y referencias
- Lista Simplemente Enlazada (Ordenada y No Ordenada)
- Lista Doblemente Enlazada
- Recursividad

TEMA 5: Pilas y Colas [4+2 horas]

- Pila
- Cola (Simple, Circular y Con Prioridad)

TEMA 6: Árbol binario [6+4 horas]

- Terminología
- Árbol Binario
- Recorridos (Pre-Order, In-Order, Post-Order)

TEMA 7: Métodos Avanzados de Ordenación [2+4 horas]

- No Recursivos (Shell)
- Recursivos (Mergesort y Quicksort)

TEMA 8: Tabla Hash [2+2 horas]

- Ventajas e Inconvenientes
- Función Hash
- Colisiones

TEMA 9: Grafos [4+2 horas]

- Terminología
- Representación en memoria (Matricial y Dinámica)
- Recorrido (En Profundidad y Por Niveles)
- Algoritmo de Dijkstra

Objetivos

- Conocer las principales estructuras de datos y saber elegir la más idónea para cada problema.
- Diseñar bibliotecas, aprendiendo a separar la especificación del cuerpo.
- Verificar las bibliotecas diseñadas mediante el diseño de baterías de pruebas y manejo de excepciones.
- Desarrollar nuevas bibliotecas mediante la reutilización de plantillas.

Metodología

- Clases magistrales en las que se introducirán los conceptos fundamentales.
- La primera clase de teoría de cada semana empieza con la solución de la práctica de la semana anterior.
- Propuesta y resolución teórica de ejercicios tipos que utilicen los conceptos explicados.
- Clases prácticas que permitan el desarrollo y ejecución de estos ejercicios teóricos y de otros nuevos propuestos como prácticas.
- Material de la asignatura accesible en el Campus Virtual de la ULPGC.

Criterios de Evaluación

Actividades que liberan materia:

- Las prácticas con el 30% de la nota final como máximo (según se indica en la descripción de las prácticas, la primera y la última no puntúan, la segunda y la cuarta puntúan un 30% de la nota final de prácticas, y la tercera puntúa el 40% restante de la nota final de prácticas). En las consideraciones generales se explica detalladamente como se realiza la evaluación de cada práctica.

Actividades que no liberan materia:

- Ninguna

Consideraciones generales:

- Al final del curso se realiza un examen teórico donde el alumno puede obtener hasta el 70% de la nota final.

- La evaluación de cada práctica se realiza en el laboratorio de forma continua. Cada práctica se divide en varios ejercicios, unos evaluados en cada clase práctica y otros evaluados en conjunto al final de la práctica. En la primera hora de cada clase práctica el alumno realiza un ejercicio asociado a cada semana. En la segunda hora se realiza una evaluación de la misma.
- El 50% de la nota de cada práctica se consigue con los ejercicios semanales y el 50% restante se obtiene con la evaluación global de la práctica. Los ejercicios semanales asociados a una práctica tienen la misma puntuación. Por ejemplo, si una práctica consta de tres ejercicios semanales cada uno puntúa 1/3 del 50% de la nota total de esa práctica.
- Un suspenso en una evaluación global de una práctica se puede recuperar en el laboratorio durante la realización de la última práctica mediante un ejercicio adicional a realizar en el laboratorio.
- Para aprobar las prácticas es necesario sacar al menos el 50% de la nota de prácticas.
- Para aprobar la asignatura no es necesario aprobar ambas partes: teórica y práctica.
- En cualquier convocatoria, los alumnos pueden obtener el 70% de la nota final (asociada a la parte teórica) realizando un examen escrito y, aquellos que no hayan aprobado las prácticas, pueden obtener el 30% (asociada a la parte práctica) realizando otro examen escrito. El alumno debe superar el 50% de la nota final para aprobar la asignatura.

Descripción de las Prácticas

Laboratorio de prácticas: Laboratorio de Programación (Nave 1A) del Dpto. de Ingeniería Telemática.

PRÁCTICA 0: Paquetes (No puntúa)

- Duración : 2 Horas
- Objetivos : Familiarizar al alumnos con el laboratorio y con las herramientas utilizadas. Introducir el concepto de paquete.

PRÁCTICA 1: Arrays (30% de la nota final de prácticas)

- Duración: 8 horas
- Objetivos: Introducción en el diseño y verificación de paquetes. Programación con clases abstractas de datos. Utilización de los arrays como estructura de datos estática. Aplicación de algoritmos básicos de ordenación.

PRÁCTICA 2: Listas (40% de la nota final de prácticas)

- Duración: 10 horas
- Objetivos: Introducción en el diseño y verificación de paquetes genéricos. Reutilización de algoritmos mediante derivación de estructuras de datos. Utilización de las listas como estructura de datos dinámica. Comparación de listas simples, dobles, con cabecera y sin cabecera. Introducción en la recursividad. Aplicación a las listas y ampliación de los algoritmos de ordenación.

PRÁCTICA 3: Árboles (30% de la nota final de prácticas)

- Duración: 6 horas
- Objetivos: Uso de la recursividad como mejor técnica para hacer recorridos completos de estructuras de datos recursivas. Utilización de los árboles como estructura de datos dinámica. Comparación entre vectores, listas y árboles.

PRÁCTICA 4: Grafos (no puntua)

- Duración: 4 horas

- Objetivos: Mostrar la utilidad de los grafos para representar estructuras reales: redes de ordenadores, mapa de carreteras, ...

Bibliografía

[1 Básico] Software construction and data structures with Ada 95.

Feldman, Michael B.

Addison-Wesley,, Reading (Massachusetts) : (1997)

0201887959

[2 Básico] Data structures and algorithms: an object-oriented approach using Ada 95 /

John Beidler.

Springer-Verlag,, Berlin ; New York : (1996)

0-387-94834-1

[3 Básico] Ada 95: the craft of object-oriented programming /

John English.

Prentice Hall,, London : (1997)

0-13-230350-7

[4 Recomendado] Programming in ADA 95 /

John Barnes.

Addison-Wesley,, Harlow (England) : (1998) - (2nd ed.)

0-201-34293-6

[5 Recomendado] Programación en C ++: algoritmos, estructuras de datos y objetos /

Luis Joyanes Aguilar.

McGraw-Hill,, Madrid : (2002)

8448124871

[6 Recomendado] Ada 95: problem solving and program design /

Michael B. Feldman, Elliot B. Koffman.

Addison-Wesley Publishing Company,, Reading, Mass : (1999) - (3rd ed.)

0-201-36123-X

[7 Recomendado] ADA as a second language /

Norman H. Cohen.

McGraw-Hill,, New York : (1986)

*0070115893 pbk 3295**

Equipo Docente

FRANCISCO JAVIER MIRANDA GONZÁLEZ

(RESPONSABLE DE PRACTICAS)

Categoría: TITULAR DE UNIVERSIDAD

Departamento: INGENIERÍA TELEMÁTICA

Teléfono: 928451240 **Correo Electrónico:** javier.miranda@ulpgc.es

WEB Personal: <http://www.iuma.ulpgc.es/users/jmiranda>

FRANCISCO JOSÉ GUERRA SANTANA

(COORDINADOR)

Categoría: *TITULAR DE UNIVERSIDAD*

Departamento: *INGENIERÍA TELEMÁTICA*

Teléfono: 928451238 **Correo Electrónico:** *francisco.guerra@ulpgc.es*

WEB Personal: *http://www.dit.ulpgc.es/usuarios/profes/fguerra/index.html*

LUIS MIGUEL HERNÁNDEZ ACOSTA

Categoría: *PROFESOR CONTRATADO DOCTOR, TIPO 1*

Departamento: *INGENIERÍA TELEMÁTICA*

Teléfono: 928451383 **Correo Electrónico:** *luismiguel.hernandez@ulpgc.es*

WEB Personal: *http://www.dit.ulpgc.es/usuarios/profes/lhdez/index.html*

Resumen en Inglés

DESCRIPTOR:

Sorting. Recursivity. Data structures: lists, trees and graphs. Programming methodology. Programs development and its implementation. Arrays and files.

GOALS

- To know the main data structures and learn to select the most appropriate for each kind of problem.
- To write libraries
- To verify libraries by means of functional tests
- To learn to extend libraries

METHODOLOGY

- The instructor presents in class the main concepts
- The instructor proposes exercises that help the students to understand the concepts presented in class
- In the laboratory the students will program complementary exercises
- The electronic documents containing complementary material will be available in the Campus Virtual server of the ULPGC.