



UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

GUÍA DOCENTE

CURSO: 2009/10

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

Información ECTS

Créditos ECTS: 7,2

Horas de trabajo del alumno: 181,5

Horas presenciales: 90,0

- Horas teóricas (HT): 51,5
- Horas prácticas (HP): 23,0
- Horas de clases tutorizadas (HCT): 17,0
- Horas de evaluación: 1,5
- otras: 0,0

Horas no presenciales: 90,0

- trabajos tutorizados (HTT): 31,0
- actividad independiente (HAI): 59,0

Idioma en que se imparte: Español

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: Programación modular (4 horas)

- Bibliotecas de programación
- Tipos privados
- Tipos abstractos de datos
- Programación Orientada a Objetos
- Diseño y paso de pruebas a una biblioteca

TEMA 2: Vectores (12 horas)

- Herencia
- Búsqueda lineal y binaria
- Métodos simples de ordenación (burbuja, inserción y selección)
- Ficheros
- Colas y pilas usando vectores
- Diseño de aplicaciones: Gestor

TEMA 3: Listas (12 horas)

- Memoria dinámica
- Tipos limitados
- Tipos controlados
- Nodo = Control + Datos
- Gestión de nodos: Siguiendo y Enlazar
- Listas simplemente y doblemente enlazadas
- Ficheros
- Colas y pilas usando listas
- Recursividad

TEMA 4: Tablas de hash (4 horas)

- Función hash
- Colisiones
- Herencia múltiple

TEMA 5: Listas ordenadas (8 horas)

- Movimiento de elementos frente a movimiento de punteros
- Herencias sin añadir atributos
- Método de ordenación rápida
- Reutilización de código: Gestor

TEMA 6: Árboles binarios (12 horas)

- Recorridos (pre-orden, en-orden y post-orden)
- Recorrido iterativo
- Árbol equilibrado
- Punteros a procedimiento
- Bibliotecas genéricas

TEMA 7: Grafos (8 horas)

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

Objetivos

1. Objetivos conceptuales

- 1.1. Conocer metodologías y herramientas que se utilizarán en la parte teórica y práctica de programación
- 1.2. Analizar el diseño de objetos a partir de la estructura bien conocida vector.
- 1.3. Aprender a gestionar objetos: inserciones, búsquedas, métodos simples de ordenación, etc.
- 1.4. Entender la relación y dependencias entre una estructura y los elementos que se almacenan en ella.
- 1.5. Comparar vectores con listas
- 1.6. Conocer los problemas que tiene la gestión de punteros
- 1.7. Aprender a diseñar métodos recursivos
- 1.8. Comparar con vectores ordenados
- 1.9. Analizar las ventajas de las tablas hash
- 1.10. Conocer la herencia múltiple
- 1.11. Comparar vectores, listas y árboles
- 1.12. Conocer las bibliotecas genéricas

- 1.13.Comparar árboles y grafos
- 1.14.Diferenciar los tipos de recorrido de un grafo

2.Objetivos procedimentales

- 2.1.Probar las metodologías y herramientas de programación.
- 2.2.Aplicar la herencia en el diseño de nuevos objetos
- 2.3.Diferenciar entre los diferentes métodos de búsqueda y ordenación
- 2.4.Experimentar las ventajas de la programación orientada a objetos en el diseño de una aplicación
- 2.5.Reconstruir una lista cambiando sólo punteros
- 2.6.Manipular memoria dinámica de forma segura
- 2.7.Aplicar las pilas al recorrido iterativo de árboles
- 2.8.Manejar los punteros a métodos
- 2.9.Aplicar los grafos a la solución de problemas

3.Objetivos actitudinales

- 3.1.Comportarse de forma autónoma en la toma de decisiones para solucionar problemas de programación
- 3.2.Comunicar de forma oral y escrita en las pruebas objetivas semanales

Metodología

La asignatura consta de dos partes claramente diferenciadas: teoría y prácticas de laboratorio.

TEORÍA

La teoría se desarrollara combinando las clases de teoría con ejercicios escritos y prácticos por parte del alumno.

* Clases de teoría:

-Actividad del profesor: Clases expositivas combinadas con la realización de ejercicios de programación. Se combinará el uso de presentaciones y herramientas de programación en ordenador y cañón, y el uso de pizarra, todo en el aula.

- Actividad del estudiante:

o Presencial: Tomar apuntes, participar en clase planteando dudas y realizando los ejercicios propuestos.

o No presencial: Preparar apuntes, estudiar la materia, realizar ejercicios propuestos por el profesor.

PRACTICAS DE LABORATORIO

-Actividad del profesor: Asesorar al alumno para que el alumno realice el diseño, la programación y ejecución de los ejercicios propuestos dentro de la jerarquía de paquetes de biblioteca asociada a cada práctica.

-Actividad de los alumnos:

o No presencial: Diseñar los ejercicios propuestos por el profesor dentro de la jerarquía de paquetes de biblioteca asociada a cada práctica.

o Presencial: Programar e incluir los ejercicios propuestos por el profesor dentro de la jerarquía de paquetes de biblioteca asociada a cada práctica y responder a las preguntas que el profesor sobre el trabajo realizado.

Criterios de Evaluación

Actividades que liberan materia:

- Las pruebas objetivas en clase de teoría con el 70% de la nota final como máximo. En las pruebas objetivas de las semanas que van de la 2 a la 14 se puede obtener el 1% de la nota final y se puede hacer en parejas (no se puede repetir pareja en las 13 pruebas objetivas). En la prueba

objetiva de la semana 15 se puede obtener el 57% de la nota final y es una prueba individual.

- 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:

- Se utiliza evaluación continua: pruebas objetivas de teoría y práctica semanales

- En las pruebas objetivas de teoría de las semanas 2 a la 14 se puede obtener el 1% de la nota final de la asignatura

- En las pruebas objetivas de teoría la semana 15 se puede obtener el 57% de la nota final de la asignatura.

- En las pruebas objetivas de práctica se puede obtener el 30% de la nota final de la asignatura.

- 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.

- Los exámenes de convocatoria son para recuperar la parte de teoría, de práctica o ambas.

- 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.

- Cuando un alumno se presenta a un examen de convocatoria, renuncia a la nota obtenida en la evaluación continua de las partes a las que se presente (teoría o práctica)

- 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] Programming in ADA 2005 /

John Barnes.

Addison Wesley,, Harlow : (2006)

0321340787

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

John Beidler.

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

0-387-94834-1

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

John English.

Prentice Hall,, London : (1997)

0-13-230350-7

[5 Recomendado] Programming in ADA 95 /

John Barnes.

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

0-201-34293-6

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

Luis Joyanes Aguilar.

McGraw-Hill,, Madrid : (2002)

8448124871

[7 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

[8 Recomendado] ADA as a second language /*Norman H. Cohen.**McGraw-Hill,, New York : (1986)**0070115893 pbk 3295****Organización Docente de la Asignatura**

Contenidos	Horas					Competencias y Objetivos
	HT	HP	HCT	HTT	HAI	
Tema 1. Práctica 0.	4,0	2,0	0,0	0,0	6,0	1.1 y 2.1
Tema 2. Prueba objetiva de teoría. Prácticas 1.1, 1.2 y 1.3. Prueba objetiva de práctica.	10,5	4,5	3,0	6,0	9,0	1.2, 2.1 y 3.2
Tema 3. Prueba objetiva de teoría. Prácticas 1.4, 2.1 y 2.2. Prueba objetiva de práctica.	10,5	3,5	3,0	7,0	11,0	1.4, 1.5, 1.6, 1.7, 1.8, 2.4, 2.5, 2.6, 3.1 y 3.2
Tema 4. Prueba objetiva de teoría. Práctica 2.3. Prueba objetiva de práctica.	3,5	1,5	1,0	2,0	3,0	1.9, 1.10, 2.6 y 3.2
Tema 5. Prueba objetiva de teoría. Prácticas 2.4 y 2.5. Prueba objetiva de práctica.	7,0	3,0	2,0	5,0	8,0	1.4, 1.7, 1.8, 2.4, 2.6, 3.1 y 3.2
Tema 6. Prueba objetiva de teoría. Prácticas 3.1, 3.2 y 3.3. Prueba objetiva de práctica.	10,5	4,5	3,0	7,0	11,0	1.4, 1.11, 1.12, 2.4, 2.7, 2.8, 3.1 y 3.2
Tema 7. Prueba objetiva de teoría. Prácticas 4.1 y 4.2. Prueba objetiva de práctica.	5,5	3,0	3,5	4,0	11,0	1.13, 1.14, 2.9, 3.1 y 3.2

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>**JOSÉ FRANCISCO DELGADO GARCÍA****Categoría:** PROFESOR ASOCIADO LABORAL**Departamento:** INGENIERÍA TELEMÁTICA**Teléfono:** **Correo Electrónico:** jose.delgado@ulpgc.es**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.