Guía docente de la asignatura

Asignatura	PROGRAMACIÓN		
Materia	INFORMÁTICA		
Módulo	MATERIAS INSTRUMENTALES		
Titulación	GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN GRADO EN INGENIERÍA TELEMÁTICA GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN GRADO EN INGENIERÍA DE SISTEMAS ELECTRÓNICOS		
Plan	416 (I.S.T.) 417 (I.T.) 460 (I.T.T.) 483 (I.S.E.)	**Código**	40864 (I.S.T.) 40924 (I.T.) 45004 (I.T.T.) 46539 (I.S.E.)
Periodo de impartición	1er. CUATRIMESTRE	**Tipo/Carácter**	FORMACIÓN BÁSICA
Nivel/Ciclo	GRADO	**Curso**	1º
Créditos ECTS	6		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	JOSÉ FERNANDO DÍEZ HIGUERA DAVID GONZÁLEZ ORTEGA DANIEL BOTO GIRALDA		
Datos de contacto (E-mail, teléfono…)	JOSÉ FERNANDO DÍEZ HIGUERA DESPACHO: 2D079 TELÉFONO: 983423000 ext. 5562 E-MAIL: josdie@tel.uva.es DAVID GONZÁLEZ ORTEGA DESPACHO: 2D022 TELÉFONO: 983423000 ext. 5552 E-MAIL: davgon@tel.uva.es DANIEL BOTO GIRALDA DESPACHO: 2L012 TELÉFONO: 983423000 ext. 5930 E-MAIL: danbot@tel.uva.es		
Horario de tutorías	Véase www.uva.es → Centros → Campus de Valladolid → Escuela Técnica Superior de Ingenieros de Telecomunicación → Tutorías		
Departamento	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

La asignatura *Programación* pertenece a los planes de estudios de los nuevos Grados que se imparten en la Escuela Técnica Superior de Ingenieros de Telecomunicación, siendo la primera asignatura de naturaleza informática a la que se enfrentan los alumnos de la titulación.

Esta asignatura trata de introducir al alumno en las técnicas de desarrollo de programas desde un punto de vista científico y técnico. La construcción de programas *eficaces* (que hagan lo que se espera de ellos) y *eficientes* (de la mejor manera posible) no es un arte, en el que la inspiración o la habilidad del programador constituya el soporte básico. Más bien al contrario, la construcción de programas es actualmente una metodología en la cual se deben seguir procesos sistemáticos para alcanzar el objetivo.

El alumno que inicia los estudios de las técnicas de programación (en cualquier titulación) puede tener un concepto equivocado de lo que es construir software. El alumno puede pensar que hacer un programa es sentarse delante de un ordenador y empezar a escribir líneas de código. Posteriormente, mediante un proceso de prueba y error, se depura el código hasta que se piensa que el programa hace lo que se pretendía. Esta técnica puede ser válida para implementar pequeños algoritmos (con pocas líneas) pero se muestra totalmente inadecuada cuando el programador se enfrenta al desarrollo de un programa "real", donde debe adaptarse a un equipo de trabajo y ceñirse a una metodología de programación concreta, que por supuesto, no incorpora en ninguna de sus técnicas la programación por prueba y error. En vista de esta realidad, uno de los objetivos de la asignatura consiste en que el alumno se acostumbre a programar siguiendo unas normas y directrices genéricas.

En otras palabras, se trata de desterrar la expresión "lo importante es que el programa funcione", sin importar cómo se ha conseguido.

Por último, no hay que perder de vista que, una vez cursada esta asignatura, el alumno debe disponer de una herramienta que tendrá que utilizar en otras asignaturas de la titulación, y posiblemente en su vida profesional.

1.2 Relación con otras materias

Esta asignatura está especialmente relacionada con la asignatura "Fundamentos de Ordenadores y Sistemas Operativos", la cual incluye programación en lenguaje de bajo nivel (ensamblador) y en lenguaje de alto nivel con llamadas al sistema operativo, y con la asignatura "Ingeniería de Sistemas Software", en la que se aplican las técnicas y procedimientos de una metodología de desarrollo software concreta al análisis y diseño de un sistema software en el ámbito de los servicios de telecomunicación

1.3 Prerrequisitos

La asignatura Programación pertenece a los planes de estudios de los nuevos Grados que se imparten en la Escuela Técnica Superior de Ingenieros de Telecomunicación, siendo la primera asignatura de naturaleza informática a la que se enfrentan los alumnos de la titulación.

Esta asignatura trata de introducir al alumno en las técnicas de desarrollo de programas desde un punto de vista científico y técnico. La construcción de programas eficaces (que hagan lo que se espera de ellos) y eficientes (de la mejor manera posible) no es un arte, en el que la inspiración o la habilidad del programador constituya el soporte básico. Más bien al contrario, la construcción de programas es actualmente una metodología en la cual se deben seguir procesos sistemáticos para alcanzar el objetivo.

El alumno que inicia los estudios de las técnicas de programación (en cualquier titulación) puede tener un concepto equivocado de lo que es construir software. El alumno puede pensar que hacer un programa es sentarse delante de un ordenador y empezar a escribir líneas de código. Posteriormente, mediante un proceso de prueba y error, se depura el código hasta que se piensa que el programa hace lo que se pretendía. Esta técnica puede ser válida para implementar pequeños algoritmos (con pocas líneas) pero se muestra totalmente inadecuada cuando el programador se enfrenta al desarrollo de un programa "real", donde debe adaptarse a un equipo de trabajo y ceñirse a una metodología de programación concreta, que por supuesto, no incorpora en ninguna de sus técnicas la programación por prueba y error. En vista de esta realidad, uno de los objetivos de la asignatura consiste en que el alumno se acostumbre a programar siguiendo unas normas y directrices genéricas.

En otras palabras, se trata de desterrar la expresión "lo importante es que el programa funcione", sin importar cómo se ha conseguido.

Por último, no hay que perder de vista que, una vez cursada esta asignatura, el alumno debe disponer de una herramienta que tendrá que utilizar en otras asignaturas de la titulación, y posiblemente en su vida profesional.

2. Competencias

2.1 Generales

- GB1. Capacidad de razonamiento, análisis y síntesis.
- GB2. Capacidad para relacionar conceptos y adquirir una visión integrada, evitando enfoques fragmentarios.
- GB3. Capacidad de toma de decisiones en la resolución de problemas básicos de ingeniería de telecomunicación, así como identificación y formulación de los mismos.
- GB4. Capacidad para trabajar en grupo, participando de forma activa, colaborando con sus compañeros y trabajando de forma orientada al resultado conjunto, y en un entorno multilingüe.
- GB5. Conocimiento de materias básicas, científicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías.
- GBE3. Capacidad para resolver problemas con iniciativa, creatividad y razonamiento crítico.
- GC1. Capacidad de organización, planificación y gestión del tiempo.
- GC2. Capacidad para comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
- GC3. Trabajar en cualquier contexto, individual o en grupo, de aprendizaje o profesional, local o internacional, desde el respeto a los derechos fundamentales, de igualdad de sexo, raza o religión y los principios de accesibilidad universal, así como la cultura de paz.

2.2 Específicas

- B2. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

3. Objetivos

Al finalizar la asignatura el alumno deberá ser capaz de:

- Conocer los conceptos relacionados con la programación
- Aplicar las técnicas y procedimientos de una metodología de programación de un sistema software.
- Codificar y probar dicho sistema, aplicando técnicas de programación orientada a procesos y a datos.
- Codificar, poner a punto y ejecutar programas sencillos en lenguaje C/C++.
- Diseñar algoritmos sencillos basados en los esquemas de recorrido y búsqueda.
- Autoevaluar el trabajo realizado e identificar los propios errores y aspectos a mejorar.
- Aprender de forma autónoma
  - Localizar y asimilar una determinada información a partir de su referencia (Comprensión)
  - Autoevaluarse o evaluar a otros a partir de unos criterios dados (Comprensión)
  - Identificar los propios errores (Comprensión)
  - Buscar información relevante para una tarea especificada (Aplicación)
- Trabajar en equipo
  - Intercambiar información a través del correo electrónico y de los foros (Comprensión)
  - Explicar al equipo la tarea realizada y asegurarse de que todos los demás la han comprendido (Aplicación)
  - Identificar adecuadamente las tareas a realizar por el equipo, repartir equitativamente las tareas, establecer fechas de entrega, e integrar las partes (Aplicación)
  - Identificar y abordar los conflictos en el funcionamiento del equipo (Aplicación)
  - Identificar los aspectos que han ido bien y los aspectos a mejorar, relativos al funcionamiento del equipo (Aplicación)

4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	20	Estudio y trabajo autónomo individual	38
Clases prácticas de aula (A)	0	Estudio y trabajo autónomo grupal	52
Laboratorios (L)	30		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	10		
Tutorías grupales (TG)			
Evaluación			
Total presencial	**60**	**Total no presencial**	**90**

5. Bloques temáticos

Las clases teóricas se articulan en tres bloques temáticos.

Los objetivos de aprendizaje describen en detalle todo lo que el alumno va a aprender durante este curso. Es importante que el alumno los tenga presente desde el primer momento, aunque el profesor le irá recordando los objetivos que están implicados en las diferentes actividades del curso.

Hay objetivos de cuatro tipos:

- Conocimiento: Información que el alumno debe recordar
- Comprensión: el alumno debe ser capaz de aplicar un procedimiento que conduce normalmente a una solución única
- Aplicación: el alumno debe tomar decisiones, y puede haber varias decisiones válidas
- Valoración: el alumno debe ser capaz de hacer las cosas de una cierta manera, por iniciativa propia

Bloque 1: Conceptos Básicos

Carga de trabajo en créditos ECTS: 1.2

a. Contextualización y justificación

En este primer bloque se introducen los conceptos básicos sobre los diferentes métodos para el desarrollo de aplicaciones informáticas, se presenta C, el lenguaje de programación que se utilizará para la implantación de los programas desarrollados, y se proporcionan los elementos básicos de la programación estructurada.

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con los tipos de datos elementales y sus operaciones (enteros, caracteres y reales)
  - Escribir los tipos de datos elementales y las operaciones que actúan sobre ellos (Conocimiento)
  - Escribir la declaración de datos de cualquiera de los tipos elementales (Comprensión)
  - Indicar el código ASCII de cualquier carácter, con la ayuda de la tabla correspondiente (Comprensión)
- En relación con las Sentencias básicas (asignación, condicionales, iterativas y de entrada/salida):
  - Describir el funcionamiento de las sentencias básicas (asignación, `if-then-else`, `switch`, `for`, `while`, `printf`y `scanf`) (Conocimiento)
  - Predecir el resultado de una secuencia de sentencias básicas (Comprensión)
  - Codificar una tarea convenientemente especificada, utilizando la secuencia de sentencias básicas adecuada (Comprensión)
- En relación con el Entorno de desarrollo:
  - Definir los conceptos de compilación, montaje y ejecución (Conocimiento)
  - Realizar las operaciones necesarias para crear/abrir un proyecto, y añadir y eliminar elementos a un proyecto (Comprensión)
  - Realizar las operaciones necesarias para editar, compilar, montar y ejecutar un programa, y localizar las carpetas donde están los archivos generados en cada uno de los pasos (Comprensión)
  - Interpretar adecuadamente los mensajes de error de compilación, y corregir el error de compilación correspondiente (Comprensión)
  - Describir las funcionalidades básicas del depurador (Conocimiento)
  - Realizar correctamente las operaciones básicas del depurador (insertar un punto de parada, ejecutar paso a paso y visualizar valores de variables) (Comprensión)
  - Identificar y subsanar los errores de ejecución de un programa, utilizando adecuadamente el depurador (Aplicación)

c. Contenidos

- ¿Qué es un ordenador?
- ¿Cómo se representan los datos en un ordenador?
- ¿Qué es un lenguaje de programación? El lenguaje C
- ¿Cómo se hace un programa? El entorno de desarrollo

- Sentencias de Control

d. Métodos docentes

- Clase expositiva participativa
- Estudio de casos en seminario y en laboratorio
- Resolución de problemas
- Aprendizaje colaborativo

e. Plan de trabajo

Ver **Anexo I**II

f. Evaluación

Test de conocimientos mínimos

Prueba de habilidades mínimas

Informe de Diseño

g. Bibliografía básica

- Apuntes de la asignatura.
- Ver Anexo II (bibliografía)

h. Bibliografía complementaria

- Ver **Anexo II** (bibliografía)

i. Recursos necesarios

- Aula con proyector multimedia y pizarra para sesiones de discusión.
- Laboratorio de prácticas, con un ordenador por alumno, para las sesiones de laboratorio. Cada ordenador debe contar con el entorno de desarrollo para el lenguaje C/C++ que se va a utilizar.
- Plataforma educativa para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.
- Foro de discusión en línea para atender preguntas y discutir aspectos relacionados con el módulo.
- Acceso fuera de clases a laboratorios de prácticas que cuenten con el entorno de desarrollo para el lenguaje C/C++.
- Acceso al material bibliográfico recomendado.

Bloque 2: Desarrollo orientado a procesos

Carga de trabajo en créditos ECTS: 2.0

a. Contextualización y justificación

El bloque 2 se centra en el desarrollo orientado a procesos. Se presentan las características de C que se utilizarán en esta metodología, y posteriormente se plantean los métodos de análisis, diseño e implantación propios del desarrollo orientado a procesos. El bloque se completa con un caso práctico, en el que se aplican a casos reales los métodos aprendidos.

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con Procedimientos y funciones
  - Describir los conceptos de procedimiento y función, las diferencias entre ellos, y su utilidad (Conocimiento)
  - Definir los conceptos: cabecera de procedimiento o función, parámetros formales, variables locales, resultado de la función, activación del procedimiento o función, parámetros reales, paso de parámetros (Conocimiento)
  - Describir la diferencia entre paso de parámetros por valor o por referencia (Conocimiento)

- Codificar convenientemente una llamada a procedimiento o función, pasando correctamente los parámetros (Comprensión)
- Codificar en forma de procedimiento o función una tarea convenientemente especificada, estableciendo adecuadamente los parámetros necesarios (Comprensión)
- Proponer una organización en bloques (procedimientos o funciones) de una aplicación determinada (Aplicación)

• En relación con Esquemas algorítmicos (recorrido y búsqueda)

- Explicar los esquemas de recorrido y búsqueda (Conocimiento)
- Adaptar los esquemas de recorrido y búsqueda a una situación convenientemente especificada, identificando conclaridad cada uno de los elementos del esquema (Comprensión)
- Elegir el esquema adecuado para resolver un problema determinado (Aplicación)
- Aplicar por iniciativa propia los esquemas algorítmicos estudiados (Valoración)

c. Contenidos

- Funciones y procedimientos
- Punteros y Arrays
- Esquemas algorítmicos
- Análisis y Diseño orientado a procesos

d. Métodos docentes

- Clase expositiva participativa
- Estudio de casos en seminario y en laboratorio
- Resolución de problemas
- Implantación y pruebas de ejecución de programas

e. Plan de trabajo

Ver **Anexo I**

f. Evaluación

Test de conocimientos mínimos
Evaluación del prototipo 1
Prueba de habilidades mínimas

g. Bibliografía básica

- Apuntes de la asignatura.
- Ver **Anexo II** (bibliografía)

h. Bibliografía complementaria

- Ver **Anexo II** (bibliografía)

i. Recursos necesarios

- Aula con proyector multimedia y pizarra para sesiones de discusión.
- Laboratorio de prácticas, con un ordenador por alumno, para las sesiones de laboratorio. Cada ordenador debe contar con el entorno de desarrollo para el lenguaje C/C++ que se va a utilizar.
- Plataforma educativa para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.
- Foro de discusión en línea para atender preguntas y discutir aspectos relacionados con el módulo.
- Acceso fuera de clases a laboratorios de prácticas que cuenten con el entorno de desarrollo para el lenguaje C/C++.
- Acceso al material bibliográfico recomendado.

Bloque 3: Desarrollo orientado a datos

Carga de trabajo en créditos ECTS: 2.8

a. Contextualización y justificación

El bloque 3 se centra en el desarrollo orientado a datos. Se presentan las características de C que se utilizarán en esta metodología, y posteriormente se plantean los métodos de análisis, diseño e implantación propios del desarrollo orientado a datos. El bloque se completa con un caso práctico, en el que se aplican a un caso real los métodos aprendidos en este bloque y en el anterior.

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con Tipos de datos estructurados:
  - Describir las estructuras de datos fundamentales, y las operaciones típicas sobre ellas (Conocimiento)
  - Escribir la declaración de una estructura de datos convenientemente especificada (Comprensión)
  - Escribir el código necesario para acceder a un elemento o conjunto de elementos de una estructura de datos (Comprensión)
  - Elegir la estructura de datos más adecuada para una aplicación determinada (Aplicación)
- En relación con Archivos:
  - Explicar el concepto de archivo, para qué sirve, y cuáles son las operaciones típicas sobre archivos de texto y binarios (crear, abrir, leer, escribir, preguntar por fin de archivo y cerrar) (Conocimiento)
  - Escribir las sentencias necesarias para realizar las operaciones básicas con archivo de texto y binarios (`fopen`, `fscanf`, `fprintf`, `fread`, `fwrite`, `feof` y `fclose`) (Comprensión)
  - Escribir las sentencias necesarias para determinar el tipo de error que se ha producido al realizar una operación con un archivo (Comprensión)
  - Determinar el formato adecuado para almacenar los datos en un archivo de texto o en un archivo binario, en función de las necesidades de la aplicación (Aplicación)

c. Contenidos

- Tipos Estructurados de Datos en C
- Asignación Dinámica de Memoria
- Archivos
- Base de datos
- Análisis y Diseño Orientados a Datos

d. Métodos docentes

- Clase expositiva participativa
- Estudio de casos en seminario y en laboratorio
- Implantación y pruebas de ejecución de programas
- Aprendizaje colaborativo

e. Plan de trabajo

Ver Anexo I

f. Evaluación

Test de conocimientos mínimos
Evaluación del prototipo 2
Prueba de habilidades mínima

g. Bibliografía básica

- Apuntes de la asignatura.

- Ver Anexo II (bibliografía)

h. Bibliografía complementaria

- Ver Anexo III (bibliografía)

i. Recursos necesarios

- Aula con proyector multimedia y pizarra para sesiones de discusión.
- Laboratorio de prácticas, con un ordenador por alumno, para las sesiones de laboratorio. Cada ordenador debe contar con el entorno de desarrollo para el lenguaje C/C++ que se va a utilizar.
- Plataforma educativa para publicar material didáctico, guías de ejercicios, soluciones, tareas, etc.
- Foro de discusión en línea para atender preguntas y discutir aspectos relacionados con el módulo.
- Acceso fuera de clases a laboratorios de prácticas que cuenten con el entorno de desarrollo para el lenguaje C/C++.
- Acceso al material bibliográfico recomendado.

6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: Conceptos Básicos	**1.2**	**Semanas 1 a 3**
Bloque 2: Desarrollo orientado a procesos	**2.0**	**Semanas 4 a 8**
Bloque 3: Desarrollo orientado a datos	**2.8**	**Semanas 9 a 15**

7. Tabla resumen de los instrumentos, procedimientos y sistemas de evaluación/calificación

Esta tabla recoge aspectos vinculados únicamente a la calificación y no a la evaluación. Se trata, pues, de aclarar cómo va a obtenerse la "nota" en esta asignatura. A continuación se da un ejemplo.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Entregables (individuales y de equipo)	20%	Será necesario realizar adecuadamente y entregar un porcentaje prefijado de los entregables del curso. En caso contrario la calificación final en la asignatura será No Presentado (N.P.).
Conocimientos mínimos	15%	Para aprobar la asignatura será necesario superar los test de conocimientos mínimos que se realizarán a lo largo del curso, o en las convocatorias de enero y julio
Habilidades mínimas	15%	Para aprobar la asignatura será necesario demostrar las habilidades mínimas en las pruebas que se realizarán a lo largo del curso, o en la convocatorias de enero y julio
Proyecto	40%	Realización de un proyecto en lenguaje C que será realizado a lo largo del curso y coevaluado por los alumnos con la supervisión del profesor
Actitud y participación	10%	Para calificar el apartado de Actitud y participación se tendrá en cuenta, entre otros, la participación frecuente en el Foro de Consultas, la seriedad en las tareas de evaluación encomendadas, la participación en clase y la observación del comportamiento en el equipo.

8. Consideraciones finales

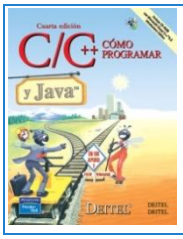
Anexo I: PLAN DE TRABAJO DE LA ASIGNATURA

Atención: El plan de trabajo que aquí se presenta es una planificación orientativa de la asignatura. Si bien el objetivo es seguir lo más fielmente posible dicha planificación, no debe entenderse como algo totalmente cerrado e inflexible, sino que puede modificarse y adaptarse si las circunstancias así lo requieren.
La planificación definitiva, con la guía de actividades de cada semana, junto con la lista y fechas de entrega de los entregables estará disponible en la página de la asignatura al comienzo del curso.

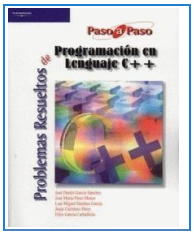
Anexo II: Bibliografía recomendada

Como se indica en la sección relativa al método docente, la asignatura tiene una gran componente de trabajo personal del alumno. La bibliografía que se propone a continuación tiene como objeto servir de soporte al aprendizaje de los conceptos y del lenguaje de programación de la asignatura, por lo que aprenderse alguno (¡o todos!) los libros no tiene sentido por sí mismo.
Sin embargo, recomendamos el uso de alguno de los libros mencionados más abajo (en el orden en el que aparecen). No es necesario adquirir más de uno, porque existen copias en la biblioteca de la escuela y porque todos ellos tratan el mismo tema, con diferentes enfoques, ejemplos y orden de exposición.

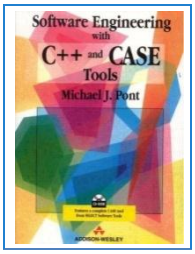
Texto Básico:



Autor	Deitel, Harvey M.
Título	Cómo programar en C/C++
Ed.	México [etc.] : Prentice-Hall, 2000

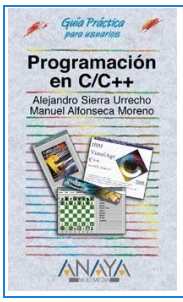


Autor	Félix García Carballeira [et al.]
Título	Problemas resueltos de programación en lenguaje C/C++
Ed.	Colección Paso a Paso. Australia [etc.] : Thomson, 2002



Autor	Michael J. Pont
Título	Software engineering with C++ and CASE tools
Ed.	Addison-Wesley 1996

Manual de bolsillo:



Autor	Alejandro Sierra Urrecho, Manuel Alfonseca Moreno
Título	Programación en C/C++
Ed.	Anaya Multimedia, 2005

Textos Complementarios

Fundamentos de Programación

	Kernighan, B. W. &Ritchie, D. M. (1992) “El lenguaje de programación C”. Ed. México. Prentice-Hall (2ª edición). Joyanes Aguilar, L. (1992). “Fundamentos de programación. Algoritmos y estructura de datos”. Ed. McGraw-Hill. Perry, G. (1998). "Aprendiendo Principios de Programación en 24 horas". Ed. Prentice-Hall.
Programación Estructurada	
	Wirth, N. (1985). “Algoritmos + Estructuras de Datos = Programas”. Ed. del Castillo. Quero Catalinas, E. & López Herranz (1997). “Programación en Lenguajes Estructurados”. Ed. Paraninfo.