



Guía docente de la asignatura

Asignatura	PROGRAMACIÓN		
Materia	INFORMÁTICA		
Módulo	MATERIAS INSTRUMENTALES		
Titulación	PROGRAMA DE ESTUDIOS CONJUNTO DE GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN Y DE GRADO EN ADMINISTRACIÓN Y DIRECCIÓN DE EMPRESAS (ITTADE)		
Plan	564	Código	45004
Periodo de impartición	1 ^{er} . 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		
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		
Horario de tutorías	Ver Tutorías en http://www.uva.es/export/sites/uva/2.docencia/2.01.grados/2.01.02.ofertaformativagrados/2.01.02.01.alfabetica/Grado-en-Ingenieria-de-Tecnologias-de-Telecomunicacion/		
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 dichos Grados.

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 del Grado, 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

El alumno que curse esta asignatura ha de poseer unos conocimientos básicos de informática a nivel usuario. En lo referente a programación, y siendo la primera asignatura de estos Grados que aborda dicha materia, se parte de la base de que el alumno no tiene conocimientos previos de la misma.



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.
- 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.
 - Autoevaluarse o evaluar a otros a partir de unos criterios dados (Evaluación).
 - Identificar los propios errores.
 - Buscar información relevante para una tarea especificada.
- Trabajar en equipo:
 - Intercambiar información a través del correo electrónico y de los foros.
 - Explicar al equipo la tarea realizada y asegurarse de que todos los demás la han comprendido.
 - Identificar adecuadamente las tareas a realizar por el equipo, repartir equitativamente las tareas, establecer fechas de entrega, e integrar las partes.
 - Identificar y abordar los conflictos en el funcionamiento del equipo.
 - Identificar los aspectos que han ido bien y los aspectos a mejorar, relativos al funcionamiento del equipo.



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	32
Clases prácticas de aula (A)	0	Trabajo autónomo grupal: proyecto integrador	58
Laboratorios (L)	40		
Prácticas externas, clínicas o de campo	0		
Seminarios (S)	0		
Tutorías grupales (TG)			
Evaluación			
Total presencial	60	Total no presencial	90



5. Bloques temáticos

Las clases teóricas se articulan en cuatro 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 presentes desde el primer momento, aunque el profesor le irá recordando los objetivos que están implicados en las diferentes actividades del curso.

5.1. Características comunes a todos los bloques

a. Métodos docentes

- Clase expositiva participativa.
- Estudio de casos en laboratorio.
- Resolución de problemas.
- Aprendizaje cooperativo.
- Aprendizaje basado en proyecto.

b. Plan de trabajo

Ver **Anexo I**

c. Evaluación

- Entregables.
- Pruebas de conocimientos y habilidades básicas.
- Evaluación de entregas parciales del Proyecto Integrador.
- Evaluación de la versión final del Proyecto Integrador.

d. Bibliografía básica

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

e. 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 los entornos de desarrollo para el lenguaje C que se van 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 aprendizaje de la asignatura.
- Acceso al material bibliográfico recomendado.

5.3. Desglose por bloques

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. El bloque se completa con casos de estudio y con la aplicación al proyecto integrador de los métodos aprendidos.

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.
 - Escribir la declaración de datos de cualquiera de los tipos elementales.
 - Indicar el código ASCII de cualquier carácter, con la ayuda de la tabla correspondiente.
- En relación con las **sentencias básicas de asignación y de entrada/salida**:
 - Describir el funcionamiento de las sentencias básicas.
 - Predecir el resultado de una secuencia de sentencias básicas.
 - Codificar una tarea convenientemente especificada, utilizando la secuencia de sentencias básicas adecuada.



- En relación con los **archivos**:
 - Explicar el concepto de archivo, para qué sirve, y cuáles son las operaciones típicas sobre archivos de texto (crear, abrir, leer, escribir, preguntar por fin de archivo y cerrar).
 - Escribir las sentencias necesarias para realizar las operaciones básicas con un archivo de texto.
 - Escribir las sentencias necesarias para determinar el tipo de error que se ha producido al realizar una operación con un archivo de texto.
- En relación con las **sentencias de control**:
 - Conocer y usar los operadores que se utilizan para construir expresiones lógicas.
 - Usar las sentencias de selección `if`, `if...else` para elegir entre varias acciones alternativas.
 - Utilizar la sentencia de selección múltiple `switch` para escoger entre muchas alternativas de acción.
 - Utilizar las etiquetas `case` para identificar las acciones alternativas en las sentencias `switch`.
 - Ejecutar sentencias repetidamente con la sentencia repetitiva `for`.
 - Usar la sentencia de repetición `while` para ejecutar sentencias repetidamente.
 - Usar la sentencia de repetición `do while` para ejecutar sentencias repetidamente.
- En relación con el **entorno de desarrollo**:
 - Definir los conceptos de compilación, construcción y ejecución.
 - Realizar las operaciones necesarias para crear/abrir un proyecto, y añadir y eliminar elementos a un proyecto.
 - 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.
 - Diseñar archivos `makefile`.
 - Interpretar adecuadamente los mensajes de error de compilación, y corregir el error de compilación correspondiente.
 - Describir las funcionalidades básicas del depurador.
 - Realizar correctamente las operaciones básicas del depurador (insertar un punto de parada, ejecutar paso a paso y visualizar valores de variables).
 - Identificar y subsanar los errores de ejecución de un programa, utilizando adecuadamente el depurador.

c. Contenidos

- Introducción a los ordenadores.
- Conceptos básicos del lenguaje C.



- Funciones de interacción con el usuario y de acceso a archivos de texto.
- Anatomía de un programa.
- Entornos de desarrollo. Depuradores.
- Sentencias de control.
- El preprocesador de C.

d. Evaluación

- Cuestionarios de Conocimientos Básicos.
- Objetivos de aprendizaje asociados a este bloque.

Bloque 2: Desarrollo Orientado a Procesos

Carga de trabajo en créditos ECTS:

1.2

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. El bloque se completa con casos de estudio y con la aplicación al proyecto integrador de los métodos aprendidos. Como resultado se obtendrá un *framework* del proyecto completo (versión 1).

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con los procedimientos y funciones:
 - Definir los conceptos de procedimiento y función, las diferencias entre ellos, y su utilidad.
 - Definir los conceptos: cabecera de función, parámetros formales, variables locales, resultado de la función, llamada a procedimiento o función, parámetros reales, paso de parámetros.
 - Describir la diferencia entre paso de parámetros por valor o por referencia.
 - Codificar convenientemente una llamada a procedimiento o función, pasando correctamente los parámetros.
 - Codificar en forma de función una tarea convenientemente especificada, estableciendo adecuadamente los parámetros necesarios.
 - Proponer una organización en bloques (funciones) de una aplicación determinada.



- En relación con los esquemas algorítmicos (recorrido y búsqueda):
 - Explicar los esquemas de recorrido y búsqueda.
 - Adaptar los esquemas de recorrido y búsqueda a una situación convenientemente especificada, identificando con claridad cada uno de los elementos del esquema.
 - Elegir el esquema adecuado para resolver un problema determinado.
 - Aplicar por iniciativa propia los esquemas algorítmicos estudiados.

c. Contenidos

- Funciones y procedimientos
- Esquemas algorítmicos.
- Programación modular.

d. Evaluación

- Evaluación de la versión 1 del proyecto integrador (*framework*).
- Cuestionarios de Conocimientos Básicos
- Evaluación de objetivos de aprendizaje conseguidos, correspondientes a este bloque.

Bloque 3: Desarrollo orientado a datos

Carga de trabajo en créditos ECTS:

2.6

a. Contextualización y justificación

El bloque 3 se centra en las estructuras de datos, tanto estáticas como dinámicas, en la creación e intercambio de datos con archivos binarios, y en la introducción a bases de datos. Se presentan las características de C que se utilizarán en esta metodología. El bloque se completa con casos de estudio y con la aplicación al proyecto integrador de los métodos aprendidos (versión 2). El *framework* realizado en la versión 1 se completa con el código necesario para generar una aplicación completa, siguiendo un esquema de programación modular basada en TADs (tipos abstractos de datos).



b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- En relación con los **tipos de datos estructurados**:
 - Describir las estructuras de datos fundamentales, y las operaciones típicas sobre ellas.
 - Escribir la declaración de una estructura de datos convenientemente especificada.
 - Escribir el código necesario para acceder a un elemento o conjunto de elementos de una estructura de datos.
 - Elegir la estructura de datos más adecuada para una aplicación determinada.
- En relación con la **asignación dinámica de memoria**:
 - Asignar y liberar memoria dinámicamente para diferentes tipos de datos.
 - Crear y redimensionar vectores dinámicos.
- En relación con los **archivos binarios**:
 - Recordar el concepto de archivo, para qué sirve, y cuáles son las operaciones típicas sobre archivos (crear, abrir, leer, escribir, preguntar por fin de archivo y cerrar).
 - Escribir las sentencias necesarias para realizar las operaciones básicas con archivos binarios.
 - Escribir las sentencias necesarias para determinar el tipo de error que se ha producido al realizar una operación con un archivo binario.
- En relación con los **vectores**:
 - Escribir la declaración de un vector unidimensional.
 - Escribir el código necesario para acceder a un elemento de un vector unidimensional.
 - Escribir la declaración de un vector multidimensional.
 - Escribir el código necesario para acceder a un elemento de un vector multidimensional.
 - Aplicar los esquemas algorítmicos de recorrido y búsqueda al caso de vectores.
- En relación a las **cadenas de caracteres**:
 - Escribir la declaración de una cadena de caracteres.
 - Escribir el código necesario para acceder a un elemento de una cadena de caracteres.
 - Utilizar las funciones de la biblioteca estándar cuyas declaraciones se encuentran en el archivo de cabecera *string.h* para manipulación de cadenas de caracteres.
- En relación a los **punteros**:
 - Comprender el concepto de puntero y sus diferencias con una variable simple.
 - Escribir el código necesario para acceder al dato apuntado por un puntero.
 - Escribir el código necesario para recorrer un vector o cadena de caracteres utilizando un puntero.
 - Implantar el paso de parámetros por dirección utilizando punteros.
- En relación a las **bases de datos**:
 - Entender el concepto de base de datos, y cómo se estructura.
 - Escribir las sentencias necesarias para crear, abrir y cerrar una base de datos sencilla (*SQLite*).
 - Escribir las sentencias necesarias para realizar las operaciones básicas sobre una base de datos (crear tablas, insertar datos, seleccionar datos).



c. Contenidos

- Tipos estructurados de datos en C.
 - Vectores estáticos y dinámicos.
 - Estructuras de datos y registros.
 - Archivos binarios.
- Asignación dinámica.
- Bases de datos.

d. Evaluación

- Evaluación de la versión 2 del proyecto integrador.
- Cuestionarios de Conocimientos Básicos correspondientes a este bloque.
- Evaluación de objetivos de aprendizaje conseguidos, correspondientes a este bloque.

Bloque 4: Introducción al desarrollo orientado a objetos

Carga de trabajo en créditos ECTS:

1.0

a. Contextualización y justificación

El bloque 4 es una breve **introducción** al desarrollo orientado a objetos. Se presentan las características básicas del lenguaje C++ que se utilizarán en esta metodología, y los conocimientos básicos para crear una clase e instanciar y manejar un objeto. El bloque se completa con casos de estudio y con la aplicación al proyecto integrador de los métodos aprendidos, convirtiendo los TADs implantados en la versión 2 en clases y objetos.

b. Objetivos de aprendizaje

Al final de este bloque el alumno será capaz de:

- Conocer los conceptos fundamentales del lenguaje C++ necesarios para trabajar con clases.
- Diseñar e implantar clases, y crear objetos a partir de ellas
- Asignar memoria dinámica utilizando los operadores de C++
- Diseñar una solución completa basada en la programación orientada a objetos, a partir de una versión previa implantada con TADs.



c. Contenidos

- Nociones básicas sobre el lenguaje C++
- Conceptos Fundamentales: Clases y Objetos.
- Encapsulamiento, Polimorfismo, Herencia

d. Evaluación

- Evaluación de la versión 3 del proyecto integrador.
- Cuestionarios de Conocimientos Básicos correspondientes a este bloque.
- Evaluación de objetivos de aprendizaje conseguidos, correspondientes a este bloque.



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	1.2	Semanas 4 a 7
Bloque 3: Desarrollo orientado a Datos	2.6	Semanas 8 a 12
Bloque 4: Introducción a la Prog. Orientada a Objetos	1.0	Semanas 13 a 14

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

A continuación se presenta la tabla que recoge los criterios de evaluación que se aplicarán en la convocatoria ordinaria.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Entregables	15%	Será necesario realizar adecuadamente y entregar un porcentaje prefijado de los entregables del curso realizados en aula y laboratorio. En caso contrario la calificación final en la asignatura será No Presentado (N.P.).
Conocimientos básicos	15%	Cada semana el alumno tendrá que realizar una prueba objetiva de conocimientos básicos sobre el tema de trabajo de dicha semana. Para superar cada prueba dispondrá de cuatro intentos durante el curso, más dos intentos adicionales en las convocatorias de enero y febrero. Es condición necesaria (pero no suficiente) para superar la asignatura que el alumno supere TODOS los conocimientos básicos.



Objetivos de Aprendizaje (insignias)	15%	<p>Será necesario conseguir un porcentaje mínimo (80%) de insignias (<i>badges</i>) para cada uno de los bloques. La insignia correspondiente a un determinado objetivo de aprendizaje se consigue al demostrar en los ejercicios realizados en el laboratorio que se ha logrado dicho objetivo de aprendizaje. Si en alguno de los bloques no se consigue dicho porcentaje, todavía se pueden conseguir en las convocatorias ordinaria y/o extraordinaria.</p> <p>Es condición necesaria (pero no suficiente) para superar la asignatura que el alumno supere el porcentaje mínimo en los cuatro bloques.</p> <p>En caso contrario la calificación final en la asignatura será No Presentado (N.P.).</p>
Proyecto integrador	40%	<p>Realización de un proyecto integrador de dificultad media, en lenguaje C/C++, que se implementará en varias versiones a lo largo del curso. Cada versión será una ampliación de la anterior, en la que se incorporarán las nuevas técnicas y conocimientos adquiridos. Los alumnos también tendrán que coevaluar el proyecto realizado por otros compañeros. Una vez entregada la última versión del proyecto, el alumno tendrá que realizar una ampliación individual del mismo. La nota final del proyecto será el producto de la media ponderada de las versiones realizadas, por la nota de la ampliación individual.</p> <p>Es condición necesaria (pero no suficiente) para superar la asignatura obtener una calificación mínima de 5 sobre 10 puntos en el proyecto, y una calificación mínima de 0.5 sobre 1 en la prueba de ampliación individual del proyecto. La nota resultante no debe ser inferior a 4</p>
Evaluación	15%	<p>La calificación de este apartado vendrá dada por las notas obtenidas por el alumno en las actividades de coevaluación y autoevaluación de ejercicios de laboratorio y de las versiones del proyecto integrador.</p>

Si un alumno no alcanza los requisitos mínimos descritos en la tabla anterior, su calificación final en la asignatura será el mínimo entre el valor calculado según la ponderación descrita en la tabla y 4.



a. Evaluación en la convocatoria extraordinaria

Esta tabla recoge los criterios de evaluación que se aplicarán en la convocatoria extraordinaria de febrero.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Conocimientos básicos	25%	Se conserva la nota de la convocatoria ordinaria, salvo que el alumno no haya superado todos los conocimientos básicos. En este último caso, el alumno solamente tendrá que recuperar los conocimientos básicos que tenga pendientes.
Objetivos de aprendizaje	25%	Se conserva la nota de la convocatoria ordinaria, salvo que el alumno no haya conseguido el porcentaje mínimo de insignias. En este último caso, el alumno solamente tendrá que conseguir las insignias necesarias.
Proyecto integrador	50%	<p>Se conserva la nota de la convocatoria ordinaria, salvo que el alumno no haya aprobado esta parte, y siempre que dicha nota supere la calificación final de 4 sobre 10. Se pueden dar varias situaciones:</p> <ul style="list-style-type: none">● El alumno no ha entregado la versión final del proyecto. En este caso tiene que entregar dicha versión y realizar la prueba de ampliación individual.● El alumno ha entregado todas las versiones del proyecto, pero la media ponderada no supera los 5 sobre 10 puntos: el alumno tiene que entregar la versión final del proyecto y realizar la prueba de ampliación individual.● El alumno ha entregado todas las versiones del proyecto, y la media ponderada supera los 5 sobre 10 puntos, pero no ha superado la nota mínima de 0.5 sobre 1 en la prueba de ampliación individual: el alumno tiene que realizar la prueba de ampliación individual.



Si un alumno no alcanza los requisitos mínimos descritos en la tabla anterior, su calificación final en la asignatura será el mínimo entre el valor calculado según la ponderación descrita en la tabla y 4.

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

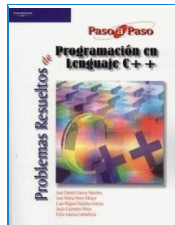
Como se indica en la sección relativa al método docente, la asignatura tiene una gran componente de trabajo personal del alumno. La bibliografía que se propone a continuación tiene como objeto servir de soporte al aprendizaje de los conceptos y del lenguaje de programación de la asignatura, por lo que 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). Existen ejemplares en la biblioteca de la escuela y todos ellos tratan el mismo tema, con diferentes enfoques, ejemplos y orden de exposición.

Texto Básico:

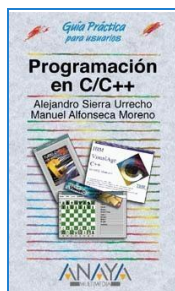


Autor	Deitel, Harvey M.
Título	Cómo programar en C/C++
Ed.	México [etc.] : Prentice-Hall, 2000



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

Manual de bolsillo:



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

Textos Complementarios

Fundamentos de Programación	
	<ul style="list-style-type: none">• 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	
	<ul style="list-style-type: none">• Wirth, N. (1985). "Algoritmos + Estructuras de Datos = Programas". Ed. del Castillo.• Quero Catalinas, E. & López Herranz (1997). "Programación en Lenguajes Estructurados". Ed. Paraninfo.